Реферат

Кузнецова Е.Ф. Управление разработкой проекта выпускной квалификационной работы: Разработка ИС организации и планирования ресурсов и производства на предприятии ТПЖА.090302.014 ПЗ: Выпускная квалификационная работа / ВятГУ, каф. САУ; рук. В.Н. Поздин. – Киров, 2019. ПЗ 163 с., 36 рис., 20 табл., 13 источников, 3 прил.

ЛОГИСТИКА, СИСТЕМА MRP, ОРГАНИЗАЦИЯ И ПЛАНИРОВАНИЕ РЕСУРСОВ, АНАЛОГИ, ВКР, УПРАВЛЕНИЕ ПРОЕКТОМ, ПЛАНИРОВАНИЕ ЗАДАЧ, СТРУКТУРА ПРОЕКТА.

Объект исследования и разработки – информационная система организации и планирования ресурсов и производства на предприятии.

Цель работы – проектирование и разработка автоматизированной системы закупок материально-технических ресурсов на предприятии.

Был выполнен анализ предметной области, определены цель и задачи проекта, выполнено планирование проекта, а именно определены сроки выполнения проекта, выполнены расчеты по окупаемости проекта. Были осуществлены разработки части программной реализации с описанием алгоритма функционирования и основных функций, описания пользовательского интерфейса. Заключительную часть составил расчет экономической эффективности разработки ИС.

Содержание

Введение.....8

Глава 1. Анализ предметной области ИС организации и планирование ресурсов и производства на предприятии. Алгоритм MRP.....10

1.1. Определение цели и задач проекта.....10

1.2. Теоретические особенности информационной системы организации и планирования ресурсов и производства на предприятии.....17

1.3. Правильная организация и этапы производственного планирования информационной системы.....21

1.4. Принцип функционирования MRP-систем при организации и планирование ресурсов и производства на предприятии.....23

1.5. Методы определения материально-технических запасов при функционировании MRP.....29

1.6. Выбор аналогов и выбор пути решений.....35

Глава 2. Анализ управления запасами на предприятии ООО «АвтоЛитМаш» 41

2.1 Финансово-экономическая характеристика предприятия.....41

2.2 Управление запасами с учетом методов логистики.....46

Глава 3. Разработка производственной программы управления запасами на предприятии ООО «АвтоЛитМаш».....53

3.1. Проектирование и реализация программы ИС.....53

3.2. Организационно-функциональная структура системы.....55

3.3. Общие сведения о проекте.....57

3.4. Назначение, цели создание системы.....58

3.5. Программное обеспечение.....58

3.6. Основные технические решения.....60

3.7. Проектирование модели предметной области.....61

3.8. Разработка и описание пользовательского интерфейса.....69

3.9 Разработка инструкции по эксплуатации ИС.....77

3.10. Структура ИС.....78
3.11. Расчёт производственной программы.....78
Глава 4. Пример расчета производственной программы.....86
4.1. Составление производственного плана и общего плана деятельности87
4.2. Планирование потребностей в материалах.....90
4.3. Планирование потребности в производственных мощностях.....92
Заключение.....107
Приложение A.....109
Приложение B.....110
Приложение C.....112
Приложение D.....114
Приложение E.....115
Приложение F.....117

Введение

Каждая современная организация, работающая в условиях рынка, осуществляет производство и сбыт продукции с целью удовлетворения потребностей рынка и получения прибыли. Важным этапом для деятельности любой организации является его план, которым необходимо руководствоваться на всех этапах функционирования предприятия. Грамотно составленный план проекта является гарантом качества его реализации, который определяет успех деятельности организации, конкурентоспособность на рынках товаров и услуг. Организация должна уметь периодически вносить поправки в свои производственные, научно-технические и сбытовые планы, управляя распоряжаться своими финансовыми и умственными ресурсами. Базой для таких возможностей являются точные, своевременные и экономически обоснованные расчеты технико-экономических показателей работы предприятия. Для выполнения этих расчетов необходимы прикладные и фундаментальные исследования. Это всё является для предприятия возможностью основательно разобраться в ходе и технико-экономическом будущем развития производства, иначе оно может потерпеть неудачу, непоправимо отстать от конкурентов. Главным показателем, характеризующим деятельность каждого предприятия, является объём в ыпуска продукции. Объём производства, а также его качественные характеристики и ассортимент занимают важное место в планировании на производстве. Данные показатели включаются в производственную программу, которая является одним из основных плановых документов любого предприятия.

Актуальность выбранной темы обусловлена тем, что через производственную программу реализуется основная цель функционирования предприятия, которая состоит в получении максимальной прибыли. Главная цель выпускной квалификационной работы – исследовать динамику и структуру производственной программы предприятия, и расчет ее основных компонентов.

Основными задачами данной работы являются представление структуры системы планирования и реализация ее основных компонентов.

«Производственная программа – это совокупное задание по выпуску и реализации продукции четкого ассортимента и качества в натуральных и стоимостных показателях, которые ориентируются на результат деятельности предприятия» [1].

Объектом исследования являются процессы планирования на предприятии и производственная программа.

Цель работы – проектирование и разработка автоматизированной информационной системы закупок материально-технических ресурсов на предприятии

Задачами работы является исследование разновидностей производственных программ, а также методы расчета в натуральном и стоимостном выражении различных показателей производственной программы.

Соответственно, результатом работы стали пример расчета производственной программы, а также исследование формирования планов и планирование на предприятии.

Глава 1. Анализ предметной области ИС организации и планирование ресурсов и производства на предприятии. Алгоритм MRP.

1.1. Определение цели и задач проекта

Целью создания выпускной квалификационной работы является проектирование и разработка автоматизированной системы закупок материально-технических ресурсов на предприятии.

В рамках данной работы изучены современные MRP системы, проведен анализ методов, моделей и средств обеспечения информационной поддержки жизненного цикла изделия в части подготовки и его производства.

Задачи:

1.Провести поиск и составить обзор методов, моделей и средств обеспечения информационной поддержки планирования МТР на предприятии

2.Провести обследование существующих систем класса MRP, логистических систем

3.Составить перечень основных функциональных задач;

4.Спроектировать контур системы

5.Разработать алгоритмы обработки данных и проведения расчетов, связанных с запасами, стратегиями поставок МТР, а также управления ими на предприятии

6.Спроектировать интерфейс, разработать и отладить программное обеспечение.

7.Подготовить пояснительную записку по ВКР

Основное назначение системы состоит в информационной поддержке отдельных этапов логистической цепи на предприятии: планирование, закупка, поставка, складирование, перемещение МТР.

Объектом автоматизации на предприятии будет выступать процесс планирования, поставки материально-технических ресурсов (МТР) - совокупность ресурсов, участвующих в производственном процессе и имеющих материально-вещественную форму.

Субъектами предметной области будут выступать специалисты, которые будут работать непосредственно с автоматизированной системой. АС осуществляет информационную поддержку закупки, поставки и движения МТР на предприятии.

Материальный поток – имеющая вещественную форму продукция, рассматриваемая в процессе приложения к ней различных логистических операций в заданном временном интервале. [3]

Формой существования материального потока может быть грузовой поток. Когда поток отнесен к моменту времени, он переходит в свою противоположность – запасы. Например, грузовой поток, отнесенный не ко временному интервалу, а к моменту времени, является запасом в пути. Материальный поток бывает внешним и внутренним (по отношению к данной логистической системе).

Запасы – материальная продукция, ожидающая вступления в процесс потребления производственного или личного или в процесс продажи.[9]

Информационная поддержка поставок производства продукции начинается с потребности в материально-технических ресурсах, планировании.

Под потребностью в материально-технических ресурсах понимается их количество, необходимое для обеспечения изготовления заданного объема продукции.

Планирование — это разработка и установление руководством предприятия системы количественных и качественных показателей его развития, в которых определяются темпы, пропорции и тенденции развития данного предприятия, как в текущем периоде, так и на перспективу.

Виды планирования на предприятии [9]:

- Текущее (годовое) планирование
- Оперативно-производственное планирование
- Директивное планирование
- Индикативное планирование
- Перспективное планирование
- Стратегическое планирование
- Тактическое планирование

Планирование направлено на оптимизацию всех видов затрат, координацию действий коллектива, предвидение событий с целью снижения риска и необоснованных потерь и готовность к быстрому реагированию на изменение окружающей среды.

Существует несколько методов планирования, такие как:

- Балансовый метод

Составление балансов по основным видам ресурсов, учёт которых возможен в натуральных и стоимостных показателях (материальные, финансовые, трудовые и др.).

- Нормативный метод

Применение обоснованных норм и нормативов использования каждого вида ресурсов в процессе производственной деятельности предприятия.

- Программно-целевой метод

Построение дерева целей, формирование программ/процедур по их реализации и определение ресурсов с учётом существующих ограничений.

- Метод планирования по технико-экономическим факторам

Предполагает учёт совокупности факторов:
- технических и технологических;
- изменение состава и объёма производимой продукции;
- совершенствование организации производства и труда;
- инфляция и т.п.

- Экономико-математическое моделирование.

Основой планирования являются объем планируемого производства, номенклатура используемых в производстве материалов и технически обоснованные нормы расхода этих материалов в расчете на единицу готовой продукции. При этом учитывается предполагаемое изменение остатков материальных ресурсов на складах предприятия.

В зависимости от учета имеющихся на предприятии запасов материалов различают брутто- и нетто-потребности в материалах.

Брутто-потребность - это потребность в материалах на плановый период без учета запасов на складе или в производстве.

Нетто-потребность - потребность в материалах на плановый период с учетом наличных запасов. Она определяется как разность между брутто-потребностью и наличными запасами материалов на данный момент времени.

Существуют три модели планирования - повторяющаяся покупка без изменений, новая покупка и повторяющаяся покупка с изменениями.

Повторяющаяся (обычная) покупка без изменений предполагает повторение ранее принятых решений по приобретению ресурсов как в части самого предмета обмена. Этот вид закупки не требует дополнительной организации деятельности службы МТО производственного предприятия: изменения структуры, численности и квалификации персонала, обеспечения и сложившегося распределения функциональных обязанностей, ответственности и взаимоотношений в процессе деятельности по обеспечению фирмы ресурсами.

Новая покупка связана с принципиальными изменениями как в области самого предмета закупки (ресурса), так и в области организации коммерческой деятельности по обеспечению предприятия. Содержание этих изменений обусловливается как изменениями стратегии и политики предприятия в целом, так и изменениями в области его коммерческой деятельности по сбыту и, соответственно, в производственно-технологической деятельности.

Повторяющаяся покупка с изменениями является неким промежуточным видом, не связанным с принципиальными изменениями, но, тем не менее, предусматривающим таковые как в части предмета покупки, так и в части соответствующей организации самого процесса. Изменения могут касаться технических, эксплуатационных и других характеристик ресурсов; структуры ресурсов; цен; условий покупки, поставок и платежа; структуры и состава поставщиков; объемов заказов и ритмичности обеспечения и др.

Определив свою потребность в материально-технических ресурсах, предприятие должно решить вопрос об источниках покрытия этой потребности. Обладая, необходимой информацией об имеющихся на рынке сырье и материалах,

их качественных характеристиках, соответствии требованиям покупателя, совершить закупку материалов.

Закупка – процесс обеспечения предприятия всеми видами материально-технических ресурсов в требуемые сроки и в объемах, необходимых для нормального осуществления его производственно-хозяйственной деятельности.

Формы снабжения предприятия

Основной задачей предприятия по организации и управлению материально-техническим обеспечением является своевременное, бесперебойное и комплектное снабжение производства всеми необходимыми материальными ресурсами для осуществления производственного процесса в точном соответствии с утвержденными плановыми заданиями.

В практике работы предприятий различают две формы снабжения [10]: транзитную и складскую.

При транзитной форме снабжения предприятие получает сырье и материалы непосредственно от предприятий, их добывающих, облагораживающих или производящих. Применение этой формы экономически оправдано во всех случаях, когда необходимое на данный промежуток времени количество сырья и материалов равно транзитной норме или больше ее.

Складская форма снабжения - необходимые материальные ресурсы предприятие получает с баз и складов снабженческо-сбытовых организаций. Данную форму снабжения экономически целесообразно использовать для

материалов, потребляемых в небольших количествах.

Цель снабжения - гарантировать, чтобы организация получила необходимые по качеству и количеству сырье, материалы, товары и услуги в нужное время, в нужном месте, от надежного поставщика, своевременно выполняющего свои обязательства, с хорошим сервисом (как до осуществления продажи, так и после нее) и по выгодной цене.

Организация снабжения предприятия ресурсами может иметь три варианта: централизованные, децентрализованные и смешанные закупки.

Централизованный вариант характерен тем, что все функции снабжения сосредоточиваются в одном отделе. Этот вариант обладает рядом преимуществ. Облегчается контроль за выполнением обязательств по закупкам, снижаются общие издержки на снабженческий процесс (транспортные, складские расходы, затраты на размещение заказов). Централизация снабженческих функций позволяет экономить средства на исследованиях рынка и всегда иметь достоверные сведения о его состоянии.

Децентрализация снабжения предусматривает самостоятельное осуществление закупок служащими производственных подразделений. При этом на закупки тратится меньше времени, так как отсутствует делегирование полномочий.

Смешанные закупки характерны для крупных предприятий с многочисленными направлениями производства, территориально удаленными друг от друга и выпускающими различные виды продукции. В этом случае производственные подразделения самостоятельно приобретают необходимые им материальные ресурсы, а отдел материально-технического обеспечения на уровне предприятия вырабатывает единую политику закупок, координируя и

контролируя процесс, осуществляет объемные закупки.

1.2. Теоретические особенности информационной системы организации и планирования ресурсов и производства на предприятии

«Согласно ст. 5 Федерального закона от 06.12.2011 № 402-ФЗ «О бухгалтерском учете» среди объектов бухгалтерского учета дифференцируются активы экономического субъекта. Активы, являющиеся конечным результатом производственного цикла, законченные обработкой и комплектацией, признаются готовой продукцией» [2]. Полуфабрикаты собственного производства, предназначенные для последующей продажи; работы и услуги промышленного характера, которые представлены на сторону, также относятся к готовой продукции. «В действующих на данный момент Методических указаниях по бухгалтерскому учету материально-производственных запасов, утвержденных приказом Министерства финансов Российской Федерации от 28.12.2001 № 119н, готовая продукция определяется как часть материально-производственных запасов (МПЗ), предназначенных для продажи, технические и качественные характеристики которых соответствуют условиям договора или требованиям иных документов, установленных законодательством. Готовая продукция – это изделия или полуфабрикаты, полностью законченные обработкой, соответствующие действующим стандартам или утвержденным техническим условиям, принятые на склад или заказчиком» [3]. Готовая продукция, которая не отвечает каким-либо ранее приведённым условиям, учитывается в составе незавершенного производства.

Одной из важнейших частей системы управления на предприятии является планирование. В настоящий момент существуют различные толкования термина «внутрифирменное планирование». Большинство авторов выделяют два подхода для определения его сущности:

а) общеэкономический – рассматривает планирование как механизм вместо рыночного взаимодействия, как комплекс управленческих решений для внутренней среды предприятия;

б) управленческий – как функцию менеджмента, определяющую цели и результаты деятельности предприятия, а также ресурсы для их приобретения. Однако в современных условиях планирование рассматривают как форму координации осуществления разнообразных вариантов деятельности вследствие общественного разделения труда. При этом из-за высокой степени неопределенности внешней среды, присущей рыночной экономике, планирование выступает как прогнозирование, ориентированное на ключевую фигуру для предприятия – потребителя. Обобщение практического опыта показывает, что планирование является частью стратегии предприятия, определяющей глобальную цель развития и частные решения для ее достижения, что обеспечивает качественное управление любой организацией в целом. Задачи планирования, актуальные на данный момент, изображены на рис. 1.1



Рисунок 1.1 – Задачи внутрифирменного планирования

«В современных условиях внутрифирменное планирование предприятия имеет перед собой две основополагающих цели, а именно:

a) определение основных направлений и пропорций развития производства с учетом материальных источников его реализации и спроса рынка;

b) повышение эффективности хозяйственной деятельности путем обеспечения оптимального использования возможностей и всех видов ресурсов предприятия» [4]. Принципы организации процесса внутрифирменного планирования на предприятии рассмотрены на рис. 1.2

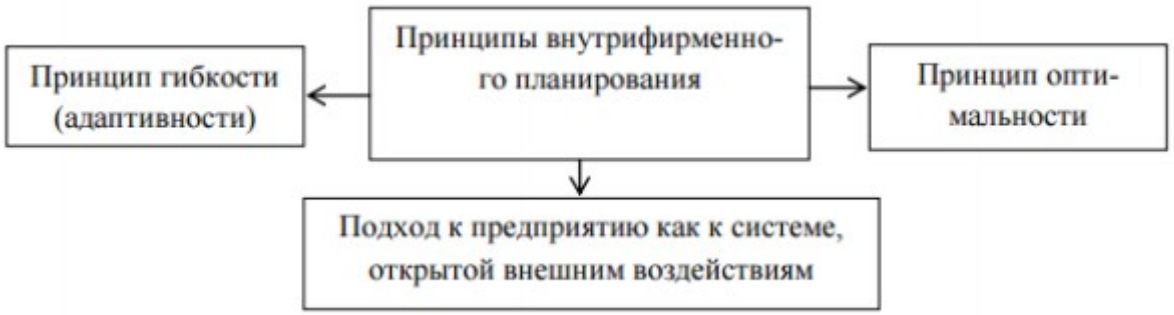


Рисунок 1.2 – Принципы организации внутрифирменного планирования

«В соответствии с принципом гибкости предприятию требуется постоянное поддержание соответствия планов изменяющимся условиям функционирования. Оптимальность планирования предполагает вариантную проработку плановых решений и выбор варианта, в наибольшей степени отвечающего интересам предприятия на основе принятых критериев. Наконец, последний принцип внутрифирменного планирования говорит о том, что предприятие - открытая система, испытывающая на себе влияние внешней среды и, в свою очередь, оказывающая на нее влияние. В процессе планирования выявляются ключевые факторы внешней среды предприятия, прогнозируется их изменение в перспективе и на этой основе разрабатывается стратегия развития предприятия» [5]. Таким образом, внутрифирменное планирование – это не просто образование планов

разных уровней, но и стратегическое управление интегрированным набором планов предприятия с учетом разных периодов, масштабов и подразделений. Можно выделить значительное количество видов планов по различным признакам и критериям, что отражено на рисунке 1.3.

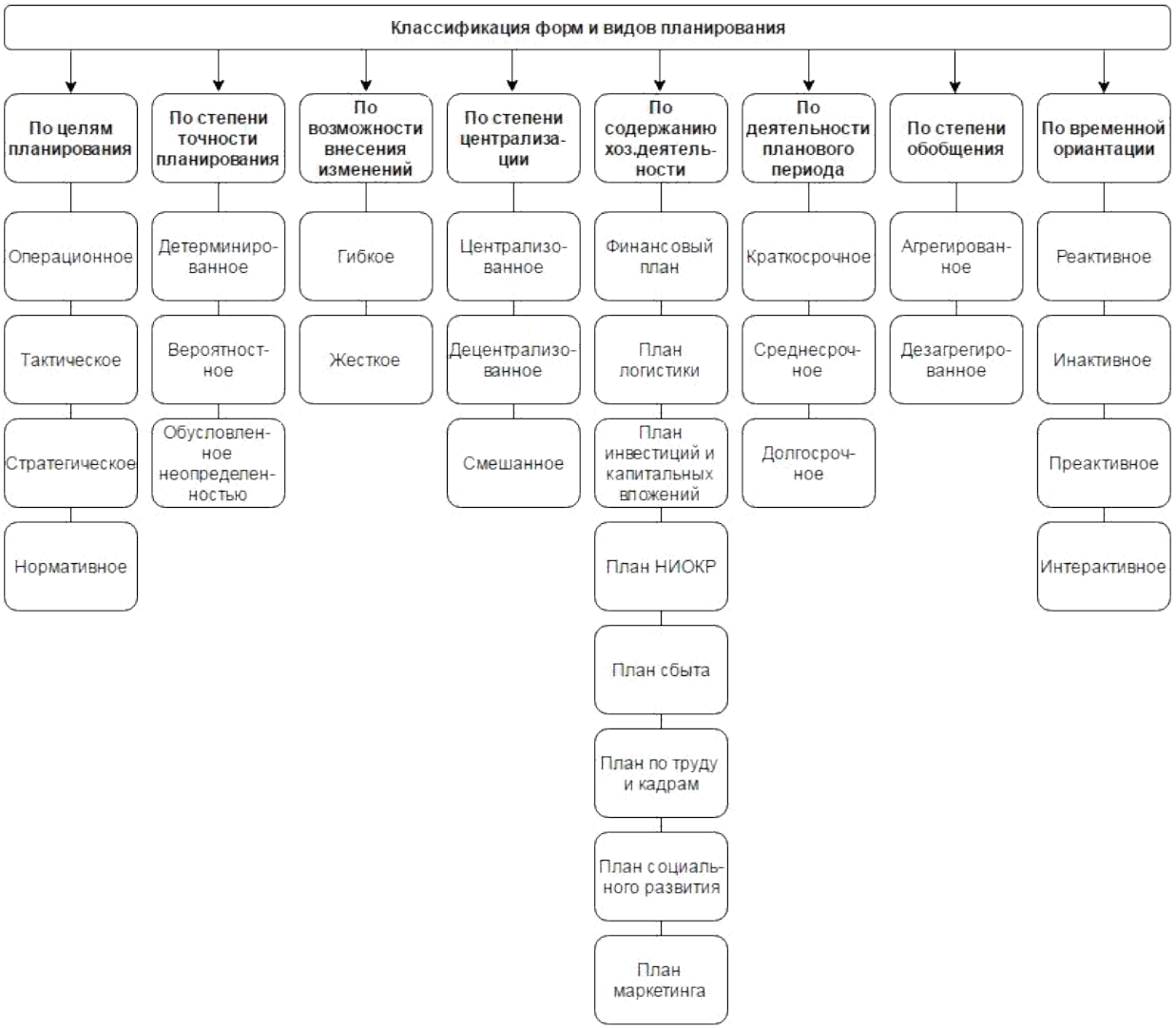


Рисунок 1.3 – Классификация форм и видов планирования на предприятии

«Любой вид плана (объемного, календарного и оперативного) разрабатывается с учетом трех составляющих его компонент. Первая компонента отражает фактическое состояние незавершенного производства по участку, цеху или предприятию в целом. Оно должно быть учтено достаточно точно при составлении планов. В составе второй компоненты находят отражение цели

(критерии), во имя которых и составляется план. Третья компонента предназначена для отражения уровня достигнутой экономической эффективности от реализации плана. Эта эффективность обеспечивается посредством экономически наиболее целесообразного использования всех видов ресурсов: живого труда, производственного оборудования, площадей и финансовых средств» [6]. Составление в качестве итогового документа внутрифирменного планирования планового баланса является чрезвычайно важным. Плановый баланс – сопоставление экономико-финансового состояния предприятия и его обязательств. Помимо этого, необходим также план результатов, в котором будут находится расчетные данные по экономическим, финансовым, производственным, трудовым, социальным, экологическим показателям.

1.3. Правильная организация и этапы производственного планирования информационной системы

«На сегодняшний день в области планирования предприятия применяют два подхода, так называемые системы «выталкивающего» и «вытягивающего» планирования. Выталкивающая система планирования производства – система планирования, позволяющая планировать организацию движения материальных потоков через производственную систему, при которой материальные ресурсы подаются с предыдущей операции на последующую, в соответствии с заранее сформированным жестким графиком поставок» [7]. «Вытягивающая» система – это система, используемая при планировании и управлении производством информацию об объемах создаваемых операционных заделов (запасов), которые определенным образом ограничены» [8]. На основе большого количества источников был проведен анализ особенностей, специфики и внедрения систем планирования для различных видов производства, и, в конечном счете, была разработана сравнительная характеристика двух систем планирования, применимых к дискретному типу производства.

№ п/п	Критерий сравнения	«Выталкивающая» система	«Вытягивающая» система
1	График поставок материальных ресурсов	Сформирован жесткий график поставок	График движения материальных потоков отсутствует
2	Информация о стадиях производственного процесса	Информация о том, продолжится ли обработка данного продукта на следующей стадии, игнорируется	Информация о состоянии следующей стадии производства является ключевым моментом и предоставляется как сигнал-требование на предыдущую операцию
3	Складирование материальных ресурсов	Результат производства каждой технологической операции в рамках незавершенного производства готовой продукции поставляется на склад	На склад поступают партии готовой продукции, прошедшие весь цикл производства
4	Централизация управления производственным циклом	Каждая технологическая операция происходит под контролем централизованного органа управления производством	Централизованное управление на каждой стадии производственного процесса отсутствует
5	Характер спроса на производимую продукцию	Ориентация на относительно постоянный спрос в течение длительного промежутка времени	Динамичный спрос, сильная сезонность спроса на продукцию, позаказное производство
6	Ритмичность производства	Использование постоянных значений ритма изготовления продукции	Значения ритма — переменные, носят характер, подчиненный нормативному уровню запасов (заделов)
7	Горизонт планирования и прогнозирования	Годовое планирование	Планирование и прогнозирование от 1 до 3 месяцев, вплоть до нескольких часов

Рисунок 1.4 – Сравнительная характеристика систем планирования производства предприятий

В «выталкивающей» системе, имеющей централизованное планирование, подразумевается доклад каждого производственного участка, получающего определенные задания на плановый период, о его выполнении перед централизованной системой управления предприятием. Возникновения отклонений, к примеру, поставок или спроса, тесно связаны с планами, которые должны оперативно пересматриваться, тем самым резко увеличивая трудоемкость работы по плану. «В отечественной практике этот вид планирования был до недавнего времени единственным; в условиях рыночной экономики он используется в основном на заготовительных предприятиях и предприятиях с массовым типом производства, производящих стандартизованную продукцию широкого назначения. «Выталкивающая» логистическая система является методологическим базисом для MRP и реализуется, как правило, на уровне современных ERP-систем» [7].

В свою очередь «вытягивающая» система, имеющая децентрализованное планирование, рассматривает только увеличенное централизованное планирование в среднесрочный период, то есть от одного до трех месяцев. В реальности под контролем центрального планового органа только у самого процесса выпуска готовой продукции, соответствуя спросу, происходит формировка графика сборки готовых изделий. «Эта система дает возможность выпуска широкой гаммы разнообразных изделий, в том числе на основе модулей, и тем самым удовлетворять разнообразный спрос» [9].

1.4. Принцип функционирования MRP-систем при организации и планирование ресурсов и производства на предприятии

Прежде чем перейти к рассмотрению MRP I системы, следует рассмотреть все основные применяемые системы планирования дискретного производства. Общие сведения о каждой из систем показаны на рис. 1.5

Система	Наименование	Сущность системы
MPS	Объемно-календарное планирование	В системе сначала формируется план продаж (в разрезе календарных периодов на основе прогноза и/или портфеля заказов), затем планируется производство и пополнение запасов за счет закупок, с учетом данных о запасах.
SIC	Статистическое управление запасами	Система использует статистические методы для планирования пополнения запасов. При этом подходе устанавливается некоторый минимальный уровень складских запасов, при достижении которого необходимо сделать новый заказ поставщику и пополнить запасы до определенного уровня.
MRP	Планирование потребности в материальных ресурсах	Система решает проблему формирования заказа на комплектующие для сложных изделий. Она опирается на объемно-календарный план, осуществляет планирование и управление всеми видами запасов - сырья и материалов, запасами сборок и окончательных комплектующих.
CRP	Планирование потребности в производственных мощностях	Основа планирования - план продаж и/или портфель заказов в разрезе календарных периодов, на основе которого осуществляется планирование потребности в производственных мощностях
MRP II	Планирование производственных ресурсов	Объединение систем планирования MRP, CRP и SIC в единое целое; их совместное, взаимоувязанное применение в процессе управления.
ERP	Система планирования ресурсов предприятия	Интегрированная система на базе ИТ для управления внутренними и внешними ресурсами предприятия (значимые физические активы, финансовые, материально-технические и человеческие ресурсы)
CSRP	Синхронизированное с потребителем планирование ресурсов	Система позволяет осуществлять планирование для предприятий со сложной разветвленной производственной структурой, включающей «дочерние» предприятия или производства, расположенные на удаленных площадках.

Рисунок 1.5 – Системы планирования

Альтернативой методу планирования по точке перезаказа стал метод планирования необходимых материалов, позволивший справиться со многими его изъянами. Этому есть достаточно простое объяснение: MRP не основывался на данных о потреблении в прошлом, ориентируясь при этом на будущие потребности. А на практике это выглядело так, что формирование заказа на пополнение запасов происходило только на необходимое количество и только во время действительной необходимости. Метод MRP берет за основу систему расчетов, которые используют данные основного производственного плана, то есть производственной программы, в построении которого фактический спрос на готовую продукцию – это исходная точка. После того, как появилась система MRP,

казалось, что основные проблемы производства были решены, и могли предприимчиво создаваться и продаваться компьютерные программы, которые реализуют ее незамысловатые принципы. Впрочем, в процессе последующего анализа ситуации, которая существует в мировом бизнесе и ее развития, было выяснено, что вся большая часть себестоимости продукции – это затраты не связанные напрямую с процессом и объемом производства. Из-за растущей из года в год конкуренции, конечные потребители продукции становятся все более “прихотливыми”, всё больше возрастают затраты на рекламу и маркетинг. Это всё требует изменения представлений о планировании коммерческой деятельности. Теперь необходимо не “что-то произвести, а потом стараться продать”, а “стараться произвести, то, что продается”. Следовательно, должна быть связь между маркетингом с планированием продаж и планированием производства. Опираясь на эти предпосылки, и появилась новая система корпоративного планирования – MRPI.

Планирование производственных ресурсов (MRP I) определяется как метод эффективного планирования всех ресурсов производственной компании. Это не только программная функция, но и управление навыками людей, которое требует точности базы данных и достаточных вычислительных ресурсов. Это общая концепция управления компанией для более продуктивного использования человеческих и корпоративных ресурсов. Метод планирования производственных ресурсов, или MRP I, – итог естественного развития MRP-алгоритма. В связи с тем, что предназначение MRP заключается в планировании необходимых материалов, логична идея захватить области деятельности, в зависимость от которых попадают обновление или расход материалов. Так, MRP I является планированием по MRP, которое включает функции управления складами, снабжением, продажами и производством. Использование вышеперечисленных функций делает реальным внесение в единую систему также функций учета и управления финансами из-за связи большей часть финансов с производством или запасами. «Для понятия алгоритма работы MRP I необходимо представить те начальные данные,

используемые этим методом. Начальные данные: 1. Материальные объекты для планирования (материалы, узлы, готовая продукция, инструменты и оснастка и т.д.) с указанием следующих параметров: а) время от возникновения потребности до ее реализации, т.е. время доставки для материалов и время изготовления для производимых изделий (время отражения); б) минимальная и кратная партии заказа; в) страховой запас по складам. 2. Маршрутный техпроцесс (МТП): а) МТП фактически является объединением спецификации и технологического маршрута. На основе спецификации, где приводится применяемость материалов, определяется количество материалов и узлов, необходимых для изготовления готовой продукции. По применяемости также определяется и материальная составляющая себестоимости готовой продукции; б) на основе информации о технологическом маршруте определяются: - время изготовления готовой продукции и, следовательно, время закупки и производства необходимых компонентов; -маршрут изготовления (по цехам, линиям и машинам); -загрузки производства; -стоимость рабочей силы и оборудования. 3. Описание производственной структуры: а) подразделения с указанием графика работы; б) рабочие центры (рабочие участки), линии с указанием их производительности, количества работающих машин и людей в бригаде. Операционные данные: 1. Ожидаемые расходы материалов или независимые потребности: а) заказы клиентов с указанием ассортимента, количества и даты закупки; б) прогнозы спроса по ассортиментным группам с указанием количества и даты; в) основной производственный план. 2. Ожидаемые приходы (пополнения материалов): а) заказы поставщикам с указанием даты прихода, ассортимента и количества; б) приходы из производства. 3. Текущий уровень запасов каждого материала в реальном времени. 4. Уровень незавершенного производства» [11].

MRP I система должна состоять из следующих модулей: 1) планирование развития бизнеса; 2) планирование продаж; 3) планирование потребностей в сырье и материалах;

4) планирование производства; 5) планирование производственных мощностей; 6) выполнение плана производства; 7) выполнение плана потребности в материалах; 8) осуществление обратной связи.

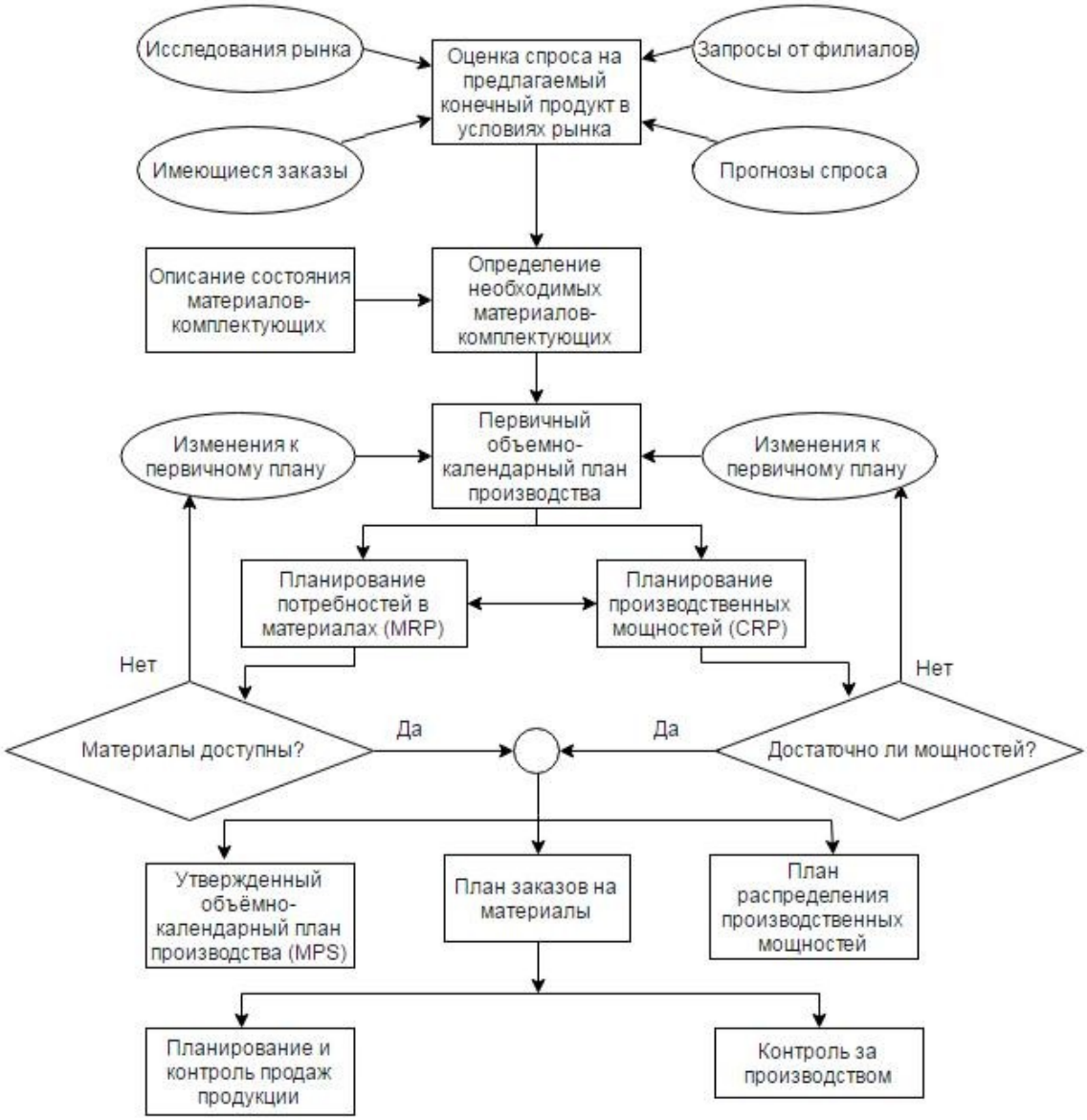


Рисунок 1.6 – Схематический план работы MRP I

Следует рассмотреть назначение каждого модуля подробнее.

- Первым необходимо отметить модуль планирования развития бизнеса, определяющий основную миссию компании (её нишу на

рынке, определение и оценку прибылей, а также финансовый ресурсы) слову, данный модуль планирования утверждает в условных финансовых единицах программу компании по производству и продаже, а также даёт оценку необходимого количества средств для инвестиций в разработку и продвижение продукта, требуемых для выхода на нужный уровень прибыли. Отсюда следует, что бизнес-план является выходным элементом данного модуля.

- Также имеет место модуль планирования продаж. Он оценивает в единицах готового изделия, какими должны быть динамика и объем продаж для выполнения установленного бизнес-плана. Преобразование плана продаж влечет за собой изменение результатов других моделей.
- Модуль планирования производства определяет программу производства всех типов готовой продукции, а также их характеристики. В пределах выпускаемой линии изделий для каждого продукта существует частная программа производства. Отсюда следует, что совокупность производственных программ для любого вида выпускаемой продукции является, в целом, производственным планом предприятия.
- Модуль планирования производственных мощностей. «Как правило, в этом модуле выполняются расчеты по определению и сравнению производственных мощностей. С небольшими изменениями этот модуль может применяться не только для производственных мощностей, но и для других видов производственных ресурсов, способных повлиять на пропускную способность предприятия. Подобные расчеты, как правило, производятся после формирования планов практически всех предыдущих уровней с целью повышения надежности системы планирования» [12].
- Стоит также рассмотреть модуль планирования потребности в материалах или типах услуг, который на базе производственный

программы для каждого вида готового продукта определяет необходимое расписание закупки и(или) внутреннего производства всех комплектующих это изделие материалов, а также их сборку.

В свою очередь, следует сказать про модуль обратной связи, позволяющий обсуждать и решать проблемы, которые могут возникнуть с дилерами, партнёрами или с поставщиками комплектующих материалов. Следовательно, с помощью данного модуля реализуется широко известный «принцип замкнутой петли» в системе. При преобразование отдельных планов, являющихся невыполнимыми и подлежащими пересмотрению, особенно необходима обратная связь.

1.5. Методы определения материально-технических запасов при функционировании MRP

Для данной работы был выбран алгоритм MRP I.

Это самый оптимальный метод для решения данной задачи.

Порядок работы (алгоритма) системы MRP I следующий:

1. «Определение независимых (входящих) потребностей».

Примеры независимых потребностей для MRPI: потребность в пополнении страхового запаса, прогнозы, основной производственный план и заказы клиентов.

2. «Расчет брутто-потребностей». Для рассмотрения данного пункта алгоритма воспользуемся примером спецификации изделия на рис.1.7

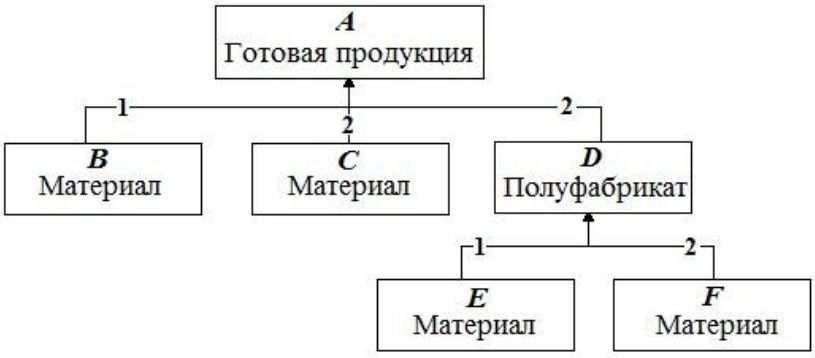


Рисунок 1.7 – Спецификация изделия

Для иллюстрации, у изделия А существует производственный план в размере 100 единиц. Используя эту информацию, состав требуемого количества компонентов будет следующий: B – 100; C – 200; D – 200; E – 200; F – 400.

3. "Расчет нетто-потребностей". Этот шаг состоит из вычисления потребностей в узлах, компонентах и материалах, с учетом уже имеющихся внезавершенном производстве или в наличии.

Таблица 1.1 – расчет нетто-потребностей

	A	B	C	D	E	F
Брутто-потребность	100	100	200	200	400	200
В наличии	0	100	80	120	280	220
Нетто-потребность	100	0	20	80	120	0

4. "Расчет нетто-потребностей во времени". Надлежащие количества вычисляются касательно всех расходов и приходов материалов. В случае выявления системой снижения уровня материала ниже конкретного уровня происходит определение необходимого количества закупки или производства для удовлетворений потребностей.

Таблица 1.2 – расчёт нетто-потребностей во времени

Дата	Компонент D	Приход/Расход	В наличии
Сегодня	В наличии	250	250
Сегодня	Расход в производство	-75	175
Завтра	Потребность 1	-50	125
Послезавтра	Потребность 2	-100	25
//**	Ожидаемый приход	200	25
//**	Потребность 3	-105	120
//**	Настоящая потребность от А	200	-80

Также имеет место расчет нетто-потребностей с учетом правила партии (минимальная партия заказа, кратность партии, периодичность заказа).

5. «Определение сроков закупки изготовления». Данный этап характеризуется определением сроков начала действий по реализации рассчитанных нетто-потребностей для отдела планирования или отдела снабжения.

Для алгоритма MRP началом является дата реализации конечной потребности, затем следует "развертка" назад во времени процесса производства продукта или закупки материала. Так, определяются даты старта производственных операций с деталями нижних уровней до самого определения дат формирования заказа поставщикам.

Таблица 1.3 – Определение сроков закупок и производства.

Компонент	Время	Примечание
A	2 дня	Изготовление из материалов B, C и узла D
B	12 дней	Время доставки материала
C	5 дней	Время доставки материала
D	2 дня	Изготовление из материалов E и F
E	2 дня	Время доставки материала
F	8 дней	Время доставки материала

Алгоритм расчёта можно проиллюстрировать при помощи схемы (рис. 1.8)

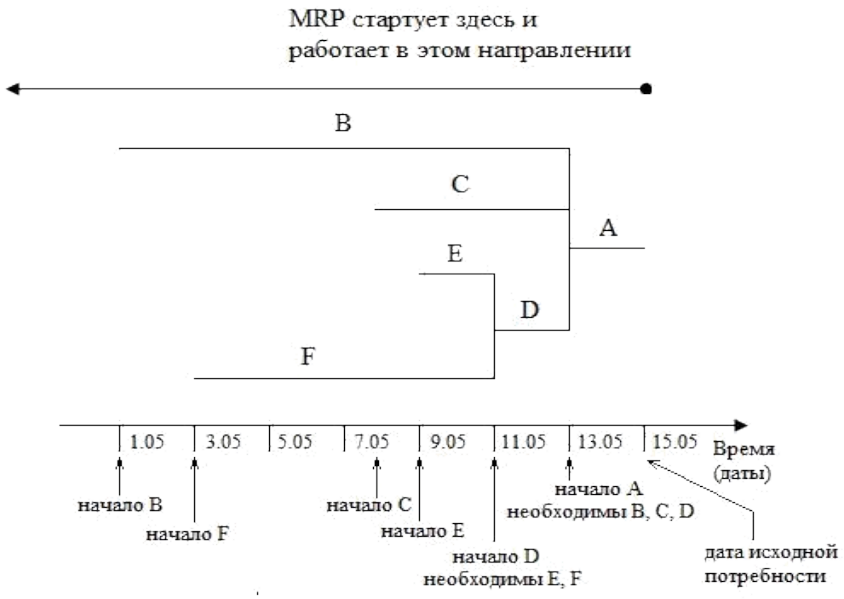


Рисунок 1.8 – Схема расчета текущих потребностей

Отрицание возможности отсутствия необходимых материалов на складе не – одна из особенностей MRP. При корректности выполнения всех исходных данных и процедуры планирования и своевременности учета отклонения в выполнении плана все поставки деталей и материалов должны быть реализованы

точно вовремя. Необходимые материалы рассчитываются, беря за основу информацию о будущих потребностях и ожидаемых уровнях запасов на складах из-за того, что MRP-метод не учитывает прошлое.

В алгоритме MRP происходит пренебрежение производственными мощностями остальных ресурсов в процедуре расчета, то есть используется принцип неограниченной загрузки. На самом деле далеко не все ресурсы предприятия считаются неограниченными. В результате еще одной функцией планирования системы, которая отличает MRP от MRP II, является планирование производственных мощностей, посредством которой производится изучение потребностей в необходимых материалах в привязке к производственным ресурсами их возможностям, а также рассматривает существующую и планируемую загрузку.

«Определение загрузки производственных мощностей и возможности реализации потребности». Определение загрузки производства происходит на основе методики производства изделия, а также сведений о производстве. На рис. 1.9 показаны шаги при исследовании загрузки и разновидности оптимизации, предполагая работу по ограниченной мощности.

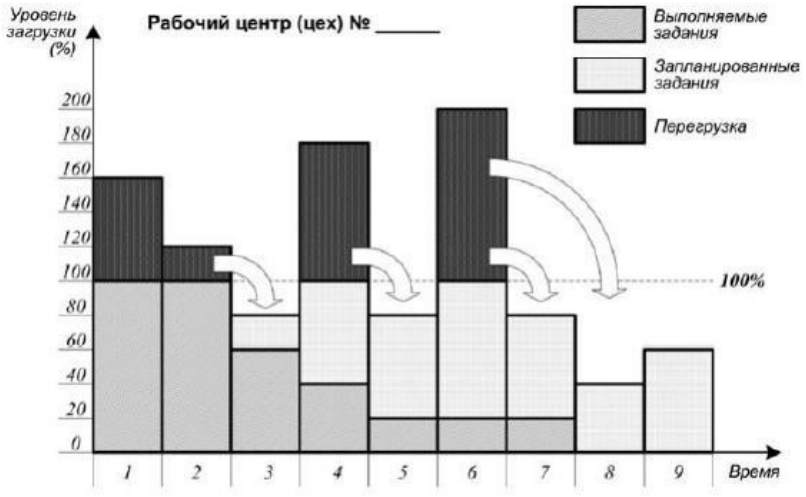


Рисунок 1.9 – Отображение загрузки рабочего центра

Существует цепь ограничений, которые встречаются в практике и снижают действенность применения данного метода. Эти ограничения следующие:

- требование абсолютной точности всех спецификаций изделий и, в случае неточности применяемости в спецификации, расчёт необходимого к закупке либо производству объема будет неверен, приведя либо к избытку, либо к нехватке;

- отсутствие ошибки в указании времени производства или закупки, иначе начало соответствующих манипуляций будет либо ранним, либо слишком поздним;

- соблюдение сроков поставки и отсутствие бракованных изделий в доставке и обязательная регистрация этого в системе вовремя, в противном случае это непременно может привести к дефициту требующихся деталей в необходимое время и так далее.

В связи получением требующихся деталей и материалов из достоверных источников прямо ко времени их необходимости, с теоретической точки зрения MRP I может считаться классической системой управления «Точно вовремя». Впрочем, с практической стороны из-за невозможности обеспечения настолько жесткого контроля исходных данных система MRP I довольно редко работает идеально. Все эти отклонения, тем не менее, могут быть сглажены наличием некоторого страхового запаса: чем больше возможность отклонения, тем больше следует делать страховой запас.

1.6. Выбор аналогов и выбор пути решений

На многих производствах имеются либо собственные системы организации и планирования ресурсов на предприятии, либо системы, которые были заказаны IT-компаниям.

Создаваемая нами ИС должна обеспечивать выполнение следующих функций:

- функций администрирования системы;
- функций менеджера;
- функций клиента.

Программа предназначена для организации работы информационной системы «МТР»

Программа должна:

1) обеспечить ввод и корректность данных:
   - Данные клиента (логин и пароль)
   - Данные Администратора
   - Данные менеджера

2) давать возможность просматривать следующую информацию:
   - По изделию;
   - По срокам изготовления;
   - По наличию на складе;
   - Отслеживание заказа;

3) обеспечивать формирование заказа:

- Проверка состояния запчастей на складе;
- Формирование срока изготовления;
- Оплата общей суммы по отделам.

Ниже рассмотрены основные системы планирования.

1) Fishbowl Manufacturing Software

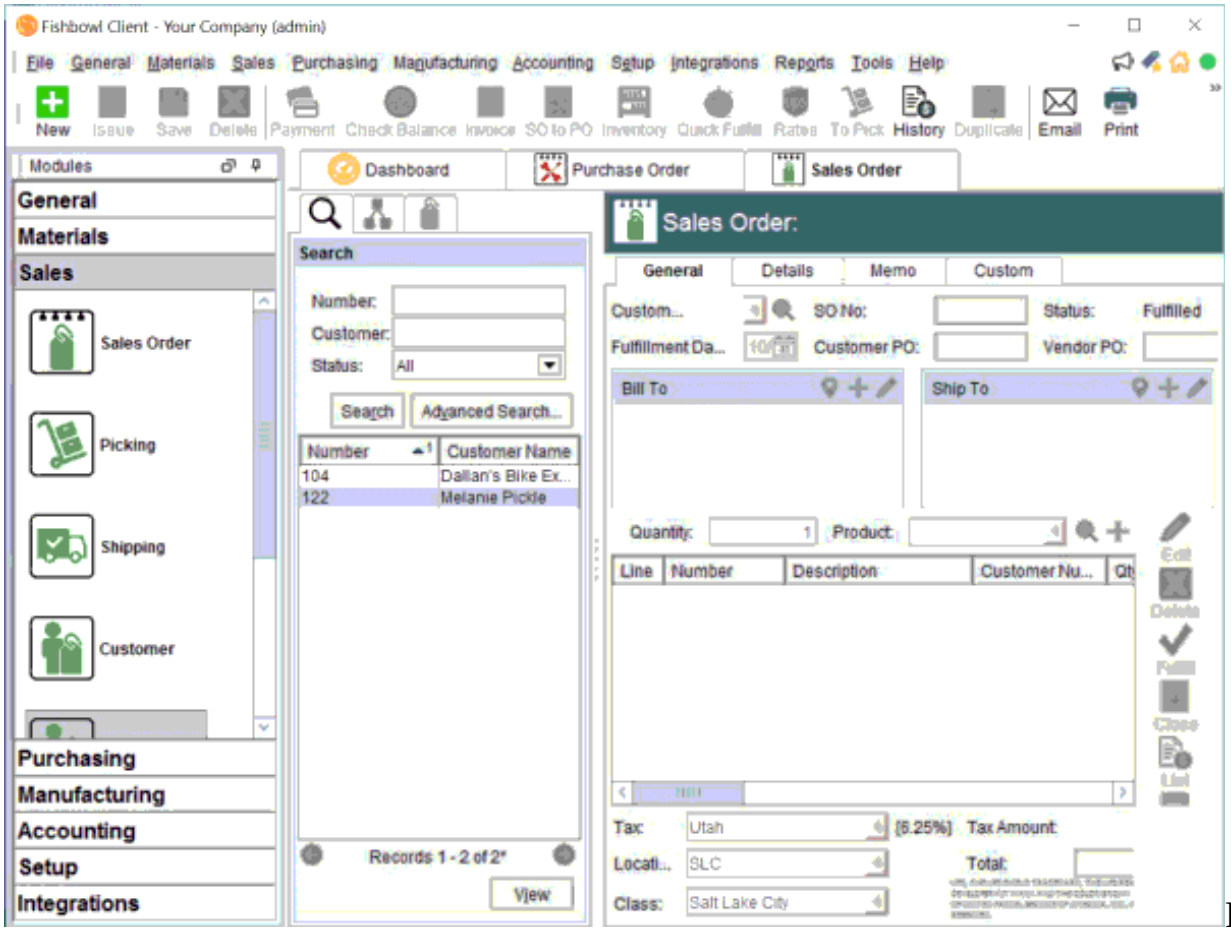


Рисунок 1.10 - Fishbowl

Fishbowl - это платформа автоматизации бизнеса и управления запасами для компаний малого и среднего бизнеса. Он интегрируется с учетными записями QuickBooks и предлагает управление запасами, планирование материальных потребностей (MRP) и контроль пола на производстве или производство.

Это система, ориентированная на инвентаризацию, с функциями для штрих-кодирования, управления активами, управления сырьевыми ресурсами, подсчета циклов и индивидуальной отчетности. Система также автоматизирует процессы цитирования, заказа и покупки. Система заказа на работу позволяет пользователям создавать и обмениваться рабочими заказами на множество задач.

Fishbowl предлагает учебные и вспомогательные ресурсы через учебные видеоролики, онлайн-уроки, собственные специалисты и многое другое. Интеграция Fishbowl с QuickBooks позволяет пользователям интегрировать процессы производства, продаж и распределения с помощью бэк-офиса. Кроме того, он также интегрируется с сторонними системами учета, доставки, электронной коммерции и торговых услуг.

2) E2 Shop System Software

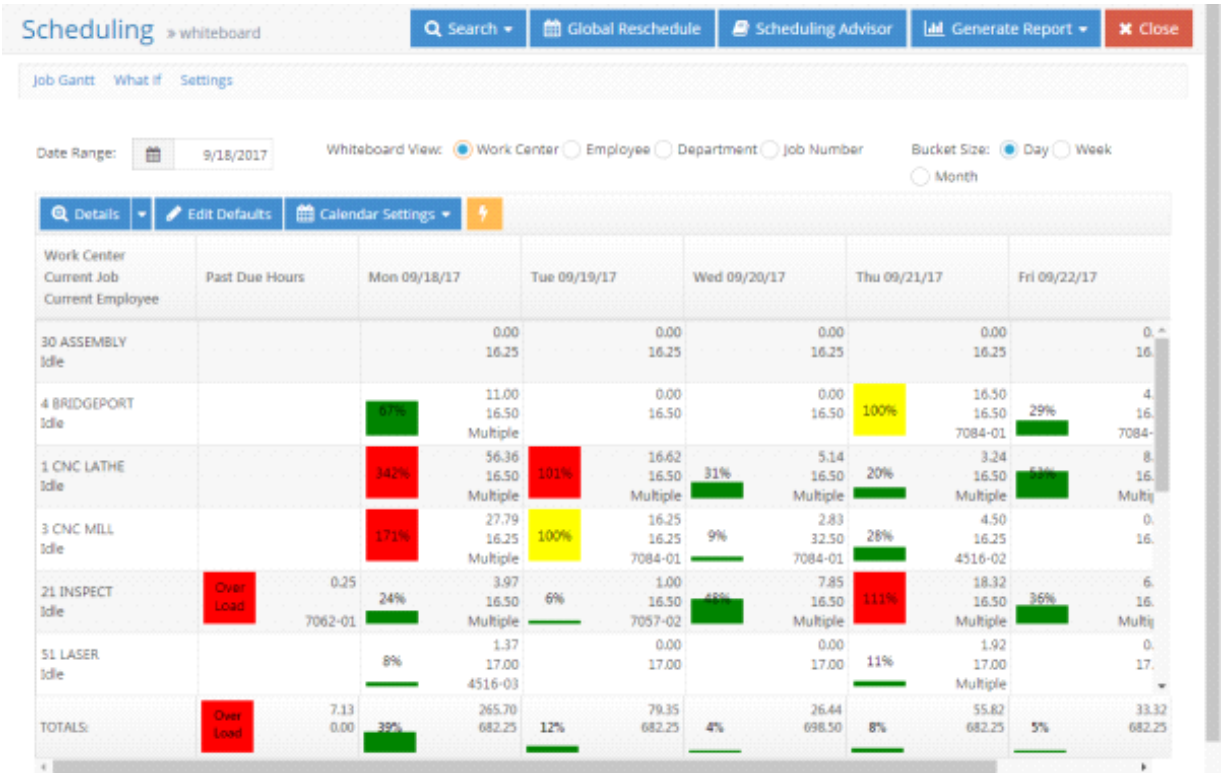


Рисунок 1.11- **E2 Shop System Software**

E2 от Shoptech Corporation - это производственное решение, которое предлагает планирование, покупку, доставку, управление клиентами и учет. Он

разработан специально для рабочих мест, контрактов и различных производителей на заказ.

Оценка E2 позволяет пользователям создавать «быстрые цитаты» и сложные оценки рабочей силы, накладных расходов, материалов и т. Д., Которые могут быть отправлены по факсу или отправлены по электронной почте с помощью прилагаемых чертежей или изображений. Эти котировки могут автоматически превращаться в заказы, без необходимости двойной записи.

E2 автоматически генерирует отгрузочные этикетки и упаковочные листы, а затем отслеживает поставки в Интернете по нескольким адресам доставки. Модуль управления материалами управляет всеми компонентами и деталями, согласовывая потребности компании в покупках с введенными заказами, а также отслеживая доставку и находив лучшие цены.

Модуль управления контактами помогает магазинам обеспечить, чтобы все остались на одной странице, от магазина до конечного клиента. Он также привлекает все финансовые показатели бизнеса, поэтому магазины могут управлять платежной ведомостью, кредиторской и дебиторской задолженностью и главной книгой из одного места, а также использовать один из 200 встроенных отчетов для всестороннего анализа.

3) NetSuite Software

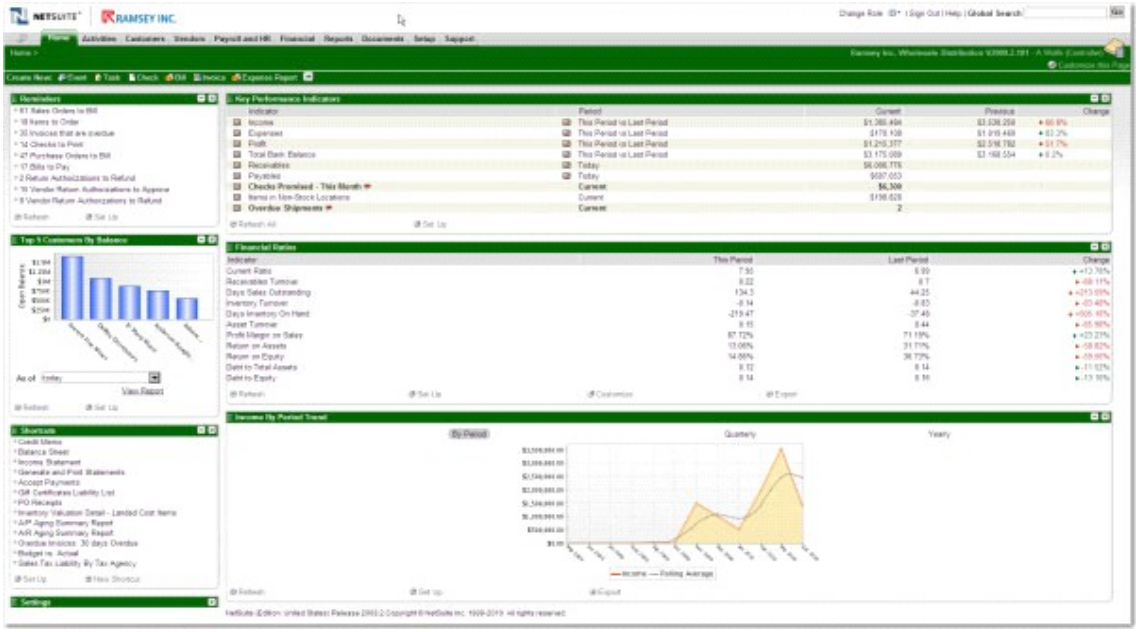


Рисунок 1.12 - NetSuite Manufacturing Edition

NetSuite Manufacturing Edition предлагает интегрированный инвентарь, управление складами, учет и финансовый менеджмент, управление заказами, управление взаимоотношениями с клиентами (CRM) и платформу электронной коммерции. NetSuite, предлагаемый как облачное решение и предоставляемый через Интернет, может использоваться производственными предприятиями для управления производственными заказами, обеспечения восстановления запасов базового инвентаря и обеспечения успеха и качества специальных заказов. Кроме того, система также поддерживает инвентаризацию для нескольких местоположений, требования к управлению сборками, спецификацию материалов, управление заказами на работу, различные методы измерения, процедуры штрихкодирования и другие необходимые бизнес-процессы в обрабатывающей промышленности. Он также может интегрировать процессы доставки с основными операторами, такими как FedEx.

NetSuite Manufacturing Edition может помочь отслеживать потенциальных клиентов и превращать их в готовые заказы, а также отслеживать связь между производством и доходами для компаний-производителей. Панель бизнес-панели

настраивается и позволяет осуществлять мониторинг бизнес-операций в реальном времени.

Как и в случае с другими пакетами SaaS, доступ к системе можно получить в Интернете с меньшими затратами на начальную стоимость и существенно менее сложным процессом внедрения.

Сделав обзор аналогов, можно заметить, что эти системы имеют высокую стоимость внедрения, что делает системы недоступными для малых и средних предприятий, очень много ненужных функций, английский язык, сложные для понимания.

Кроме того, рассмотренные выше аналоги ориентированы на решение комплекса задач. Работа системы построена заранее составленной производственной программе и установленному графику заказов на поставку МТР (один раз в две недели).

В связи с тем, что ни один из существующих аналогов для решения поставленных проблем в данном проекте не подходит, было решено разработать собственную систему планирования ресурсов «ИС ПОСТАВКА МТР». Преимуществами этой программы будут:

- Невысокая цена, а значит быстрая окупаемость
- Доступна для быстрого понимания
- Отсутствуют лишние функции, только самое основное
- Русский язык
- Доступ без Интернета
- Не требует мощную систему
- Приятный глазу простой интерфейс

Глава 2. Анализ управления запасами на предприятии ООО «АвтоЛитМаш»

2.1 Финансово-экономическая характеристика предприятия

Полное официальное наименование: Общество с ограниченной ответственностью «АвтоЛитМаш». Дата регистрации: 28 декабря 2011 г. Литейное предприятие было запущено в 2011 году в российской столице автомобилестроения, в городе Тольятти, завод специализируется на производстве крупных и мелких партий высокотехнологичных отливок, изготовлении прототипов и опытных образцов из черных и цветных металлов. В основу технологии производственных процессов предприятия входят кокильное и центробежное литьё, литьё в песчано-глинистые формы. Деятельность ООО «АвтоЛитМаш» охватывает весь комплекс производственных услуг в сфере технического и художественного литья. Общество осуществляет следующие виды деятельности в сфере производства: изготовление алюминиевого литья в кокиль и песчаные формы; художественное литьё из чугуна, бронзы, алюминия; механическая обработка литья. Производственная мощность предприятия: 350 тонн в год чугунного и стального литья в песчано– глинистые формы с размерами опок от 400х500х150/150мм до 1500х1400х500/500мм; 140 тонн отливок из меди и медных сплавов песчаноглинистые и металлические формы; 240 тонн высококачественных алюминиевых отливок, получаемых методом гравитационного центробежного литья в металлические, гипсовые и песчано-глинистые формы с использованием песчаных стержней. Основу коллектива предприятия составляют люди, прошедшие все ступени профессионального роста от литейщика или формовщика до начальника участка или руководителя предприятия. Перспективы развития ООО «АвтоЛитМаш» на 2015-2018гг. заключается в сохранении конкурентной позиции фирмы в регионе в качестве высокотехнологического предприятия машиностроительного комплекса. В основу своей стратегии руководство приняло принцип поступательного развития с усилением роли стимулирования инновационных компонентов в сфере литейных

технологий [25]. Основные направления: 1. сохранение и укрепление существующих позиций в Поволжском регионе, Самарской и Саратовской областях в частности; 2. увеличение объемов производства; 3. реализация комплекса мероприятий, направленных на повышение узнаваемости бренда «АвтоЛитМаш» в профессиональной среде; 4. совершенствование внутренних управленческих процедур; 5. совершенствование ценовой стратегии; 6. совершенствование системы риск-менеджмента; 7. проведение мероприятий по повышению квалификации работников. На рынке фирма позиционирует себя, как надежный партнер, нацеленный на установление долгосрочных взаимовыгодных хозяйственных связей. Услугами ООО «АвтоЛитМаш» пользуются такие предприятия как: ОАО «Гидроавтоматика», г. Самара; ЗАО «Джи Эм – АВТОВАЗ», г. Тольятти; ОАО «АК Туламашзавод», г. Тула; ФГУП ГНП РКЦ «ЦСКБ-Прогресс», г. Самара; ООО ПКФ «Полюс», г. Зеленодольск, Республика Татарстан; ООО ПКФ «ЭксФорма», г. Саратов; ООО НПФ «ПроАвтоКом», г. Тольятти; ООО «ТМССпорт», г. Тольятти; ООО «ФорМаш», г. Тольятти. Микросреду образует сам завод и непосредственно связанные с ним общей деятельностью контрагенты: поставщики, заказчики, конкуренты, маркетинговые посредники. ООО «АвтоЛитМаш» работает с 3 поставщиками: «ЭнергоСпецСплав», «ВолгаПромАвтоматика», SIMENS. У ООО «АвтоЛитМаш» три основных конкурента: ООО «Вершина», ООО «СамЛит» и ПГ «СОЭЗ». Финансовый посредник – Открытое акционерное общество «Газпромбанк». Банк оказывает кредитные услуги, финансирует сделки. Агентство по оказанию маркетинговых услуг – ООО «ТехноЛайф»: разрабатывала для «АвтоЛитМаш» фирменный стиль, логотип и сайт. В микросреде на литейный завод «АвтоЛитМаш» имеют большое влияние государственные учреждения. Министерство промышленности и технологий Самарской области регламентирует деятельность завода, утверждает стандарты в области производства. Главный офис и производство «АвтоЛитМаш» размещено в г. Тольятти. В целях повышения производительности и наиболее полного удовлетворения потребностей клиентов предприятием в 2013 году открыто обособленное подразделение в г.Самара – отдел

по развитию коммерческой деятельности. Предприятие имеет высокий рыночный потенциал. Несмотря на положительные финансовые результаты, на предприятии существует проблема простоя производственных мощностей. Руководство в лице директора ООО «АвтоЛитМаш» формирует цели и задачи компании, определяет общую стратегию и политику. Отдел по развитию коммерческой деятельности должен согласовывать свои планы с планами, намеченными директором, и утверждать их до начала реализации. Бухгалтерия занимается изысканием и использованием средств для финансирования плана маркетинга. Конструкторское бюро отвечает создание качественного, безопасного и выгодного для клиента проекта будущих отливок. Отдел снабжения отвечает за наличие всех необходимых материалов и оборудования. Производственные подразделения отвечают за выполнение плана производства заявленного качества, и в поставленные сроки. Отдел по развитию коммерческой деятельности заключает договора на изготовления отливок и осуществления других услуг или работ предприятия, фиксируя все сделки, информацию по которым, отправляет в бухгалтерию. Отдел технического контроля осуществляет политику руководства в области качества (рис. 2.1).

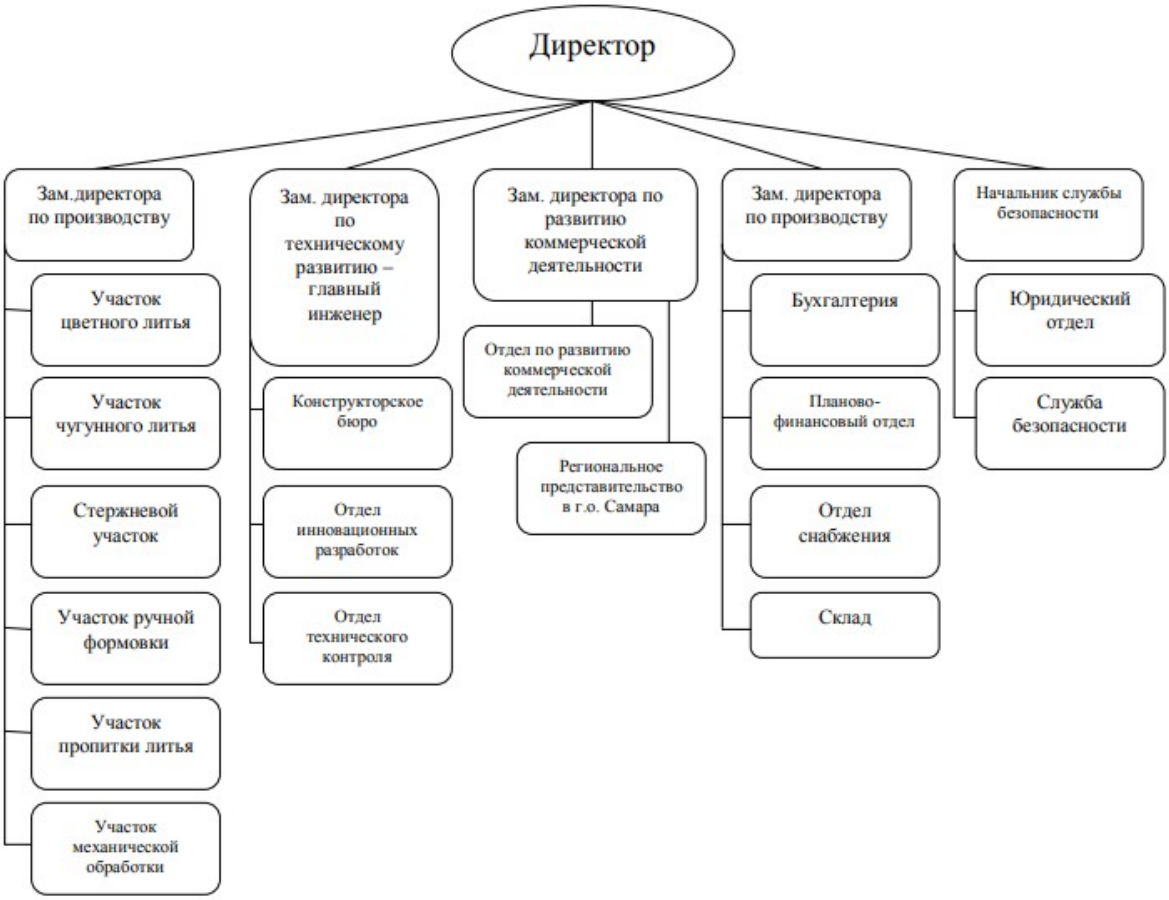


Рисунок 2.1 - Организационная структура ООО «АвтоЛитМаш»

Основной стратегической целью руководства ООО «АвтоЛитМаш» является обеспечение устойчивого роста эффективности функционирования предприятия, а также реализация концепции инновационного развития. Достижению этой цели способствует выполнение следующих задач:

1) Построение эффективной организационной структуры.

2) Устойчивое и существенное присутствие на рынке путем усиления конкурентных преимуществ за счет: ориентация производства на выполнение наиболее рентабельных и перспективных в долгосрочном периоде заказов; увеличения доли рынка в каждом стратегическом регионе; развитие системы качества;

3) Повышение финансовой эффективности за счет: стратегического планирования и контроля основных показателей деятельности; наращивание объемов производства; сокращения объема издержек.

Основная цель деятельности руководства рабочих и сотрудников ООО «АвтоЛитМаш» в области качества – производство отливок, максимально удовлетворяющей требованиям заказчиков. Для достижения этой цели предприятие определяет следующие задачи: обязательное выполнение требований, стандартов и непрерывное совершенствование системы управления качеством предприятия на базе международного стандарта ISO 9001-2000; поддерживать высокий организационный и квалификационный уровень деятельности персонала предприятия, стимулировать личную ответственность каждого сотрудника за качество продукции в рамках своей компетентности; совершенствовать средства производства, технологию и организацию работ на всех стадиях жизненного цикла изделий; обеспечивать идентификацию своих отливок с целью оперативной реакции на возможные рекламации для непрерывного совершенствования качества и удовлетворения запросы предприятия-потребителей; стремиться к высокой культуре партнерства предприятия с поставщиками и потребителями на основе взаимовыгодного сотрудничества; обеспечивать максимальное использование всех ресурсов; улучшать экономическое положение предприятия и благополучие его работников за счет повышения качества выпускаемой продукции. Укреплять корпоративную ответственность за результаты своего труда. Миссия компании ООО «АвтоЛитМаш» формулируется как обеспечение основательного подхода к каждому клиенту, к каждому этапу исполнения заказа, в области изготовления крупных и мелких партий высокотехнологичных отливок по самым выгодным ценам.

2.2 Управление запасами с учетом методов логистики

Цель формирования запасов в ООО «АвтоЛитМаш» - гарантирование нынешней производственной и сбытовой деятельности. Во время регулирования запасами они надлежащим образом группируются для обеспечения дальнейшего распределения способов управления ими. Так, классификации запасов по их видам на данном предприятии выделены: сырье, материалы и другие аналогичные ценности; затраты в незавершенном производстве; готовая продукция и товары для перепродажи; расходы будущих периодов. Главной миссией данного анализа является определение степени обеспеченности изготовления и продажи продукции подходящими запасами в предыдущем периоде и оценка эффективности их использования [18].

Анализ структуры резервов предприятия, помещенный в запасы, дает шанс найти ресурсы снижения необходимости в оборотных средствах способом оптимизации запасов, из условного снижения в сравнении с подъемом производства, ускорение расчетов или, наоборот, вовлечение средств в оборот при невыгодном положении дел [30

Таблица 2.2 – Структура запасов

Показатель	2016г. тыс.руб	2017г. тыс.руб.	2018г. тыс.руб.	Абсолютный прирост		
				2017/2016	2018/2017	2018/2016
Запасы, всего	59777	90455	98201	30678	7746	38424
сырье, материалы, аналогичные ценности	7239	9845	8067	2606	- 1778	828
затраты в незавершенном производстве	11894	17829	15134	5935	- 2695	3240
- Готовая продукция на перепродажу	40441	62461	67964	22020	5503	27523
- Расходы будущих периодов	203	320	7036	117	6716	6833

На основании анализа структуры запасов ООО «АвтоЛитМаш» можно сделать вывод, что общая величина запасов на предприятии в конце 2016 года составляла 59 777 тыс. руб., к концу 2017 года их величина возросла на 30 678 тыс. руб., а к концу 2018 года составила 98 201 тыс. руб., что говорит о снижении эффективности управления запасами на предприятии.

Запасы на сырье и материалы в 2017 году составили 7 239 тыс. руб., а в 2017 году увеличились на 2 606 тыс. руб. Однако в 2018 году запасы снизились и составили 8 067 тыс. руб.

Затраты в незавершенном производстве на протяжении рассматриваемого периода росли в абсолютном выражении. Запасы готовой продукции выросли на 27 523 тыс. руб. в 2018 году по сравнению с данными за 2016 год, доля же в общей сумме запасов оставалась почти неизменной на протяжении 2016-2018 года. Расходы будущих периодов имеют тенденцию к возрастанию с каждым отчетным годом, если в 2016 году они составили 203 тыс. руб., то в 2018 – 7 036 тыс. руб. Данная ситуация объясняется тем, что на предприятии более активно идет производство и выпуск новой продукции, что требует более крупных денежных вложений.

Таблица 2.3 – Структура запасов, %

Показатель	2016г. уд. вес, %	2017г. уд. вес, %	2018г. уд. вес, %
Запасы всего	100,00	100,00	100,00
Сырье, материалы, другие аналогичные ценности	12,11	10,88	8,22
Затраты в незавершенном производстве	19,89	19,72	15,42

Продолжение таблицы 2.3

Показатель	2016г. уд. вес, %	2017г. уд. вес, %	2018г. уд. вес, %
Готовая продукция и товары на перепродажу	67,65	69,05	69,2
Расходы будущих периодов	0,35	0,35	7,16

На основании структуры запасов в % ООО «АвтоЛитМаш» можно сделать вывод, что общая величина запасов на предприятии в конце каждого года составляет 100%.

Запасы на сырье и материалы в 2016 году составили 12,11%, но в 2017 году запасы незначительно уменьшились, что составили 10,88%. Также в 2018 году запасы снизились и составили 8,22%.

Затраты в незавершенном производстве на протяжении рассматриваемого периода росли в абсолютном выражении, но их доля в общей стоимости запасов находится примерно на одном уровне – 15 – 20%.

Запасы готовой продукции выросли на 1,55% в 2016 году по сравнению с данными за 2016 год, доля в общей сумме запасов оставалась почти неизменной на протяжении 2016 – 2016 года.

Расходы будущих периодов имеют тенденцию к возрастанию с каждым отчетным годом, если в 2016 году они составляли 0,35% от всех запасов, то в 2018 – 7,16%. Данная ситуация объясняется тем, что на предприятии более активно идет производство и выпуск новой продукции, что требует более

крупных денежных вложений.

Прежде всего, анализ запасов сырья и материалов начинают, с сопоставления их необходимого количества для производства и заказа материалов. Изучение запасов необходимых товаров проводят по каждому объекту раз в квартал. Анализ соблюдения в ООО «АвтоЛитМаш» нормы запасов на выполнение работ по кварталам представлен в таблице 2.4.

Таблица 2.4 – Запасы необходимого для производства количества сырья ООО «АвтоЛитМаш»

Дата	Плановая норма товарных запасов, тыс. руб. года			Фактические Товарные запасы, тыс. руб. года			Отклонение от норматива, тыс. руб. года		
	2016	2017	2018	2016	2017	2018	2016	2017	2018
1.01	9832	10840	13200	10000	11400	13865	+168	+560	+665
1.04	10200	13340	15740	11960	13790	15980	+1760	+450	+240
1.07	11600	9734	9023	12030	9900	9350	+430	+166	+327
1.10	18000	12700	18152	18600	13625	18550	+600	+925	+398
31.12	13300	14220	8715	14750	14773	9305	+1450	+553	+590

Таким образом, в таблице 2.4, представлен анализ сравнения нормы заказа и фактического поступления. Увеличение поступления происходит за счет того, что в течение квартала при выполнении продаж возникает необходимость в увеличении производства товаров. Следовательно, дается заявка на закупку

дополнительного сырья. В случае остатка сырья материалов после сдачи продукции, товар отправляется на склад и оттуда реализуется.

Проведем прогнозирование потребности в запасах, используя метод прогнозирования по средним значениям (таблица 2.5) [28].

Таблица 2.5 – Прогноз потребления предыдущего года на основе среднедневного потребления

Месяц	Фактические отгрузки за месяц	Число рабочих дней	Среднее потребление в день	Прогноз среднедневного потребления	Прогноз месячного потребления
Январь	16832	16	1052	0	0
Февраль	50244	19	2645	1052	19988
Март	43694	21	2081	2645	55545
Апрель	54235	22	2466	2081	45782
Май	41361	21	1970	2466	51786
Июнь	22037	22	1002	1970	43340
Июль	19750	23	859	1002	23046
Август	57289	21	2729	859	18039
Сентябрь	42418	22	1929	2729	60038
Октябрь	41217	22	1874	1929	42438
Ноябрь	29663	21	1413	1874	39354
Декабрь	12497	22	569	1413	31086

На основе таблицы 2.5 построим диаграмму для наглядного представления фактических отгрузок товара по месяцам.

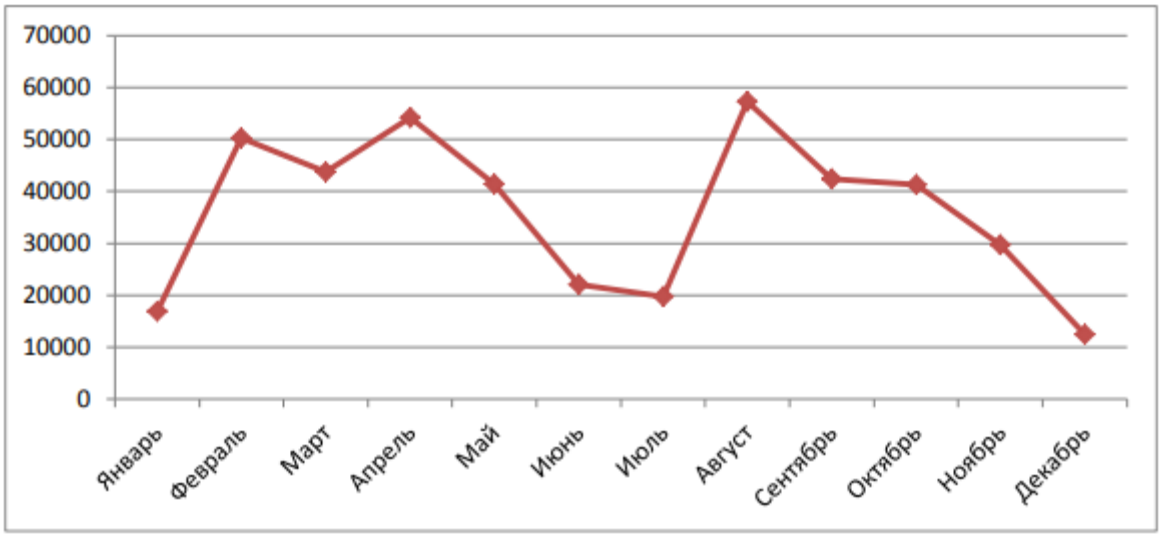


Рисунок 2.2 – Динамика фактических отгрузок товара по месяцам

А также на основе таблицы 2.5 мы построим диаграмму среднедневного потребления запаса по месяцам.

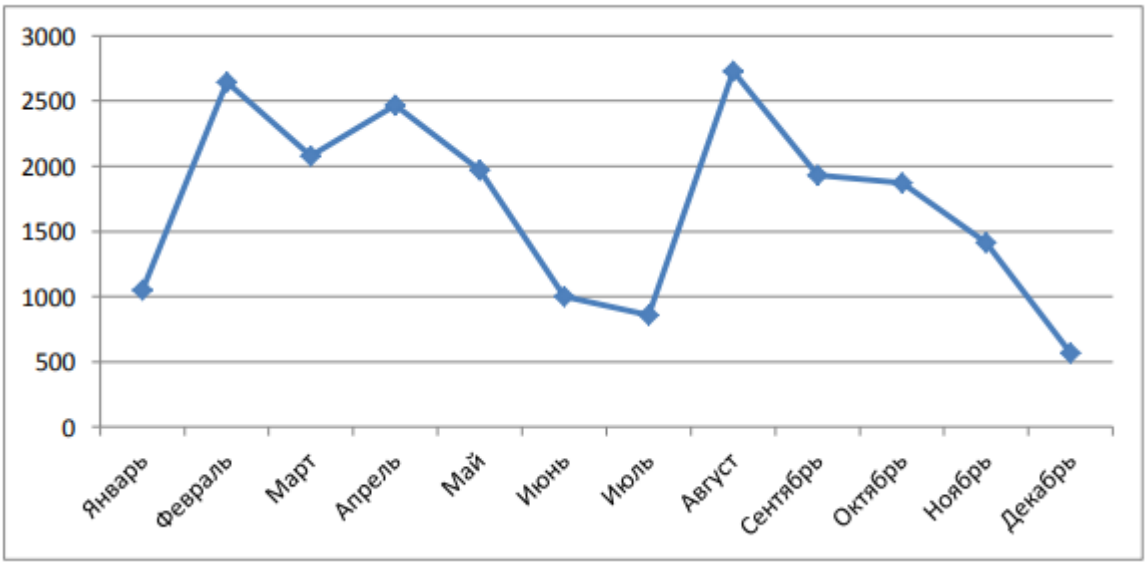


Рисунок 2.3 – Динамика среднедневного потребления запаса по месяцам

Как показал проведенный анализ стабильность потребления запасов остается нестабильной, что затрудняет планирование, в результате чего создаются их излишки.

Глава 3. Разработка производственной программы управления запасами на предприятии ООО «АвтоЛитМаш»

3.1. Проектирование и реализация программы ИС

Производственная программа разрабатывается не на всех предприятиях, которые осуществляют выпуск продукции для дальнейшей реализации потребителям или выполняют отдельные виды работ и услуг. Программа включает расчеты объема производства продукции (работ, услуг) и их обоснование производственными мощностями, материальными и трудовыми ресурсами. Производственная программа, будучи важнейшей частью бизнес-плана, показывает направление развития предприятия, толкование цели, определения профиль специализации предприятия, а также процесс развития отношений с другими предприятиями. В программе отражается величина продукции по номенклатуре, ассортименту, качеству в стоимостном и натуральном выражениях. Номенклатура — это упорядоченное многообразие товаров и услуг. В номенклатуре продукции, которая выпускается на предприятии, содержатся наименования изделий с указанием их количества, качества, сроков изготовления. Ассортимент продукции служит показателем доли отдельных ее видов в производственной программе. Качество — свойство продукции, обуславливающее её пригодность к употреблению и её способность быть использованной по назначению.



Рисунок 3.1 – Содержание понятия «производственная программа»

В ходе разработки производственной программы следует согласовать фактические и потенциальные способности предприятия относительно выпуска продукции (т.е. включая производственную мощность). Производственной мощность представляет собой максимально допустимый выпуск изделий при определённых условиях эксплуатации оборудования и производственной ресурсов в соответствующий промежуток времени.

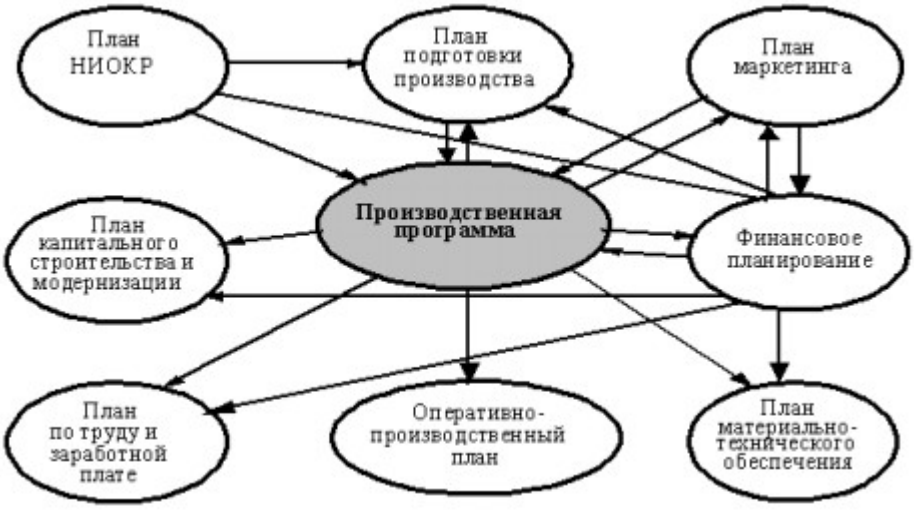


Рисунок 3.2 – Взаимосвязь производственной программы с другими планами

3.2. Организационно-функциональная структура системы

Целью данного проекта является проектирование и разработка прототипа такой информационной системы, которая подходила бы к требованиям максимального количества предприятий. Данное условие является обязательным для создания тиражируемого программного продукта. Поэтому важным является построение такой типовой модели функционирования, которая отражает предметную область целиком, без привязки к конкретному объекту.

Контур автоматизированной системы, спроектирован с помощью CASE-средства AllFusion Process Modeler, предназначенного для моделирования бизнес процессов. Использовалась методология IDEF0.

IDEF0 является методологией функционального моделирования и графической нотации. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов.

В стандарте IDEF0 наиболее важная функция находится в верхнем левом углу, также есть правило стороны:

- стрелка входа приходит всегда в левую кромку активности;
- стрелка управления — в верхнюю кромку;
- стрелка механизма — в нижнюю кромку;
- стрелка выхода — в правую кромку.

Модель содержит контекстную диаграмму декомпозиции первого уровня (рис.3.3), которая является обобщенной, и контекстную диаграмму декомпозиции второго уровня, которая уже представлена в более детализированной форме.

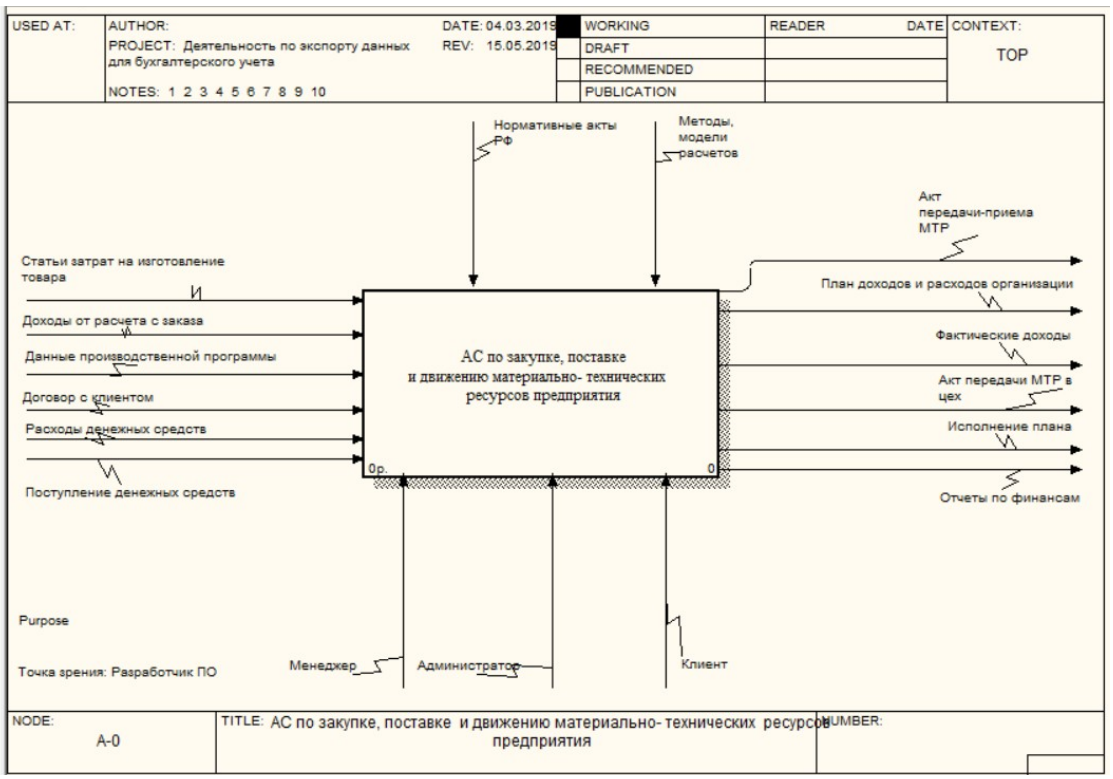


Рисунок 3.3 - Контекстная диаграмма системы IDEF0

На диаграмме первого уровня представлен один блок «ИС ПОСТАВКА МТР». Его деятельность регулируются нормативной документацией, методами и моделями расчетов.

Нормативная документация:

- технологические карты;
- договоры о поставке продукции;
- устав предприятия.

Механизмами являются пользователи Менеджер, Администратор и Клиент.

Данные производственной программы являются объектами, которые используются и преобразуются АС для получения акта передачи-приема МТР и акта передачи МТР в цех.

Для того чтобы увидеть структурно-функциональный смысл проектируемой ИС, представлен следующий уровень декомпозиции (см. рис.3.4).

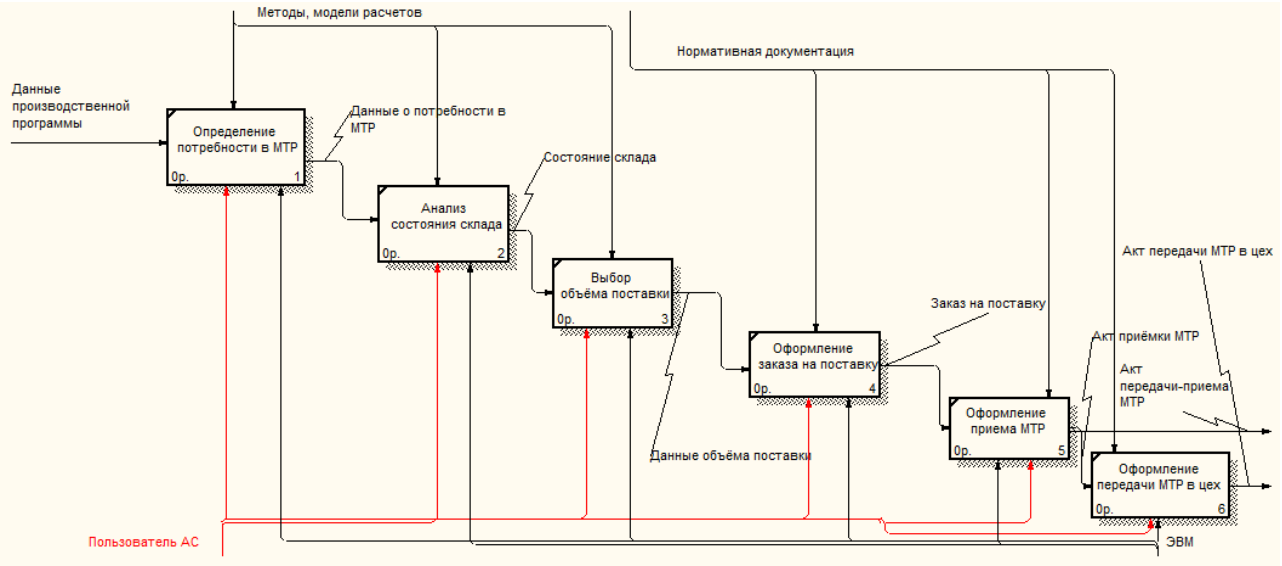


Рисунок 3.4 - Декомпозиция контекстной диаграммы деятельности по передаче данных из системы организации и планирования ресурсов и производства на предприятии в систему

3.3. Общие сведения о проекте

Полное наименование системы – автоматизированная информационная система «ИС ПОСТАВКА МТР».

Краткое (рабочее) наименование – ИС «МТР».

Руководитель проекта: Кузнецова Екатерина Федоровна.

Заказчик системы: машиностроительный завод «АвтоЛитМаш».

ИС МТР служит для планирования и производства ресурсов на предприятии.

Основанием для создания системы является проверка на удобство использования программы на производстве.

3.4. Назначение, цели создание системы

Система ИС «МТР» предназначена для:

- уменьшения трудоемкости объемных расчетов;
- для удобного и быстрого использования;
- автоматизации наиболее трудоемких и приоритетных процессов сбора, хранения и обработки информации;
- уменьшение экономических затрат

Целью разработки системы ИС «МТР» является улучшение качества предоставляемых заводом услуг за счет улучшения качества выполнения

- управленческих,
- бухгалтерских
- аналитических и других задач.

3.5. Программное обеспечение

Программное обеспечение – совокупность программ на носителях данных и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС.

В зависимости от функций, выполняемых ПО, его можно разделить на базовое (системное) и прикладное программное обеспечение.

Базовое ПО обеспечивает рабочую среду для прикладных программ.

Базовым ПО является:

- операционная система Microsoft Windows 10;
- СУБД Microsoft SQL Server 2017;

База данных системы должна обеспечивать:

I. логическую целостность данных системы при любых действиях со стороны пользователей системы;

II. логическое разделение информации для различных групп и категорий пользователей системы;

- среда программирования Microsoft Visual Studio 2017.

Система должна обеспечивать выполнение следующих функций:

- функций администрирования системы;
- функций администратора;
- функций клиента.

Далее идут требования к составу и квалификации пользователей системы.

Администратор базы данных системы. Необходимо знание основ реляционных баз данных, языка SQL, знание основ C# и основ администрирования клиент – серверных СУБД. Умение работать с проектной документацией.

Менеджер. Уверенный пользователь ПК, умение работать с Windows приложениями, отличное знание документооборота и всех тонкостей ее функционирования. Умение работать со справочной системой программных продуктов.

Клиент. Уверенный пользователь ПК, умение работать с Windows приложениями.

Ниже представлены требования к надежности системы.

Срок наполнения базы данных: от 15 лет для автоматизации одного завода, от 5 лет при создании единой информационной среды для всех подразделений.

Система должна обеспечивать защиту хранимой информации от несанкционированного доступа и порчи.

Разграничение прав пользователей по уровню доступа к данным, а также логическая целостность данных должны быть обеспечены на уровне базы данных и клиентской части.

Прикладное ПО предназначено для решения конкретных задач пользователя. Прикладным ПО является исполняемое приложение «ИС ПОСТАВКА МТР».

3.6.Основные технические решения

Техническое обеспечение – совокупность всех технических средств, используемых при функционировании ИС.

Минимальный состав технического обеспечения:

- персональный компьютер;
- манипулятор – мышь, клавиатура;
- монитор VGA 15,6” с разрешением 1366 * 768;
- источник бесперебойного питания.

Минимальные требования, предъявляемые к техническому обеспечению:

- процессор Intel Pentium IV 1,80 ГГц;
- оперативная память ёмкостью 512 Мб;
- 500 Мб свободного места на жестком диске.

Автоматизированная система имеет невысокие требования к техническому обеспечению, что позволяет использовать ее даже на маломощных системах.

3.7. Проектирование модели предметной области

Инфологическая модель

Инфологической моделью предметной области называют описание предметной области, выполненное с использованием специальных языковых средств, и не зависящее от используемых в дальнейшем программных и технических средств.[6]

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных.

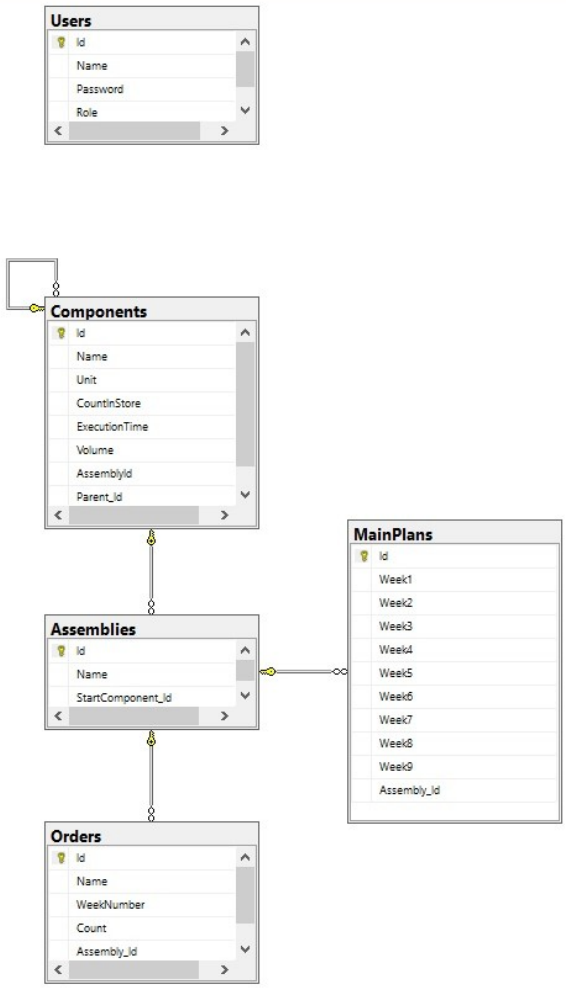


Рисунок 3.5 – Инфологическая модель

База данных состоит из следующих сущностей:

Users, Components, Assemblies, MainPlans, Orders.

Сущность Users не имеет связей, тк это отдельная таблица для авторизации пользователей.

В данном разделе выполнено графическое представление моделей логического и физического уровней нотации ERD.

Логическая модель соответствует уровню концептуального проектирования и позволяет разработать теоретическую концептуальную модель БД.

Сущности в модели описывают типы информационных объектов, хранимых и обрабатываемых в моделируемой предметной области. Каждый тип объекта характеризуется определенным набором свойств и представляется в реальном мире множеством экземпляров. Атрибуты описывают свойства сущностей. Каждый атрибут характеризует определенное свойство сущности. Для графического представления логической модели используется нотация IDEF1X. Необходимо выделить набор сущностей с их свойствами для выбранной предметной области. Полученные данные представлены в таблице.

Таблица 3.1 – Сущности и свойства

Сущность	Описание сущности	Свойства	Описание свойства
Users	Сущность содержащая информацию о данных пользователей	Id	Уникальный идентификатор
		Name	Наименование логина
		Password	Наименование пароля
		Role	Наименование роли пользователя
Components	Сущность содержащая информацию о составе изделия	Id	Уникальный идентификатор
		Name	Наименование вида затрат
		Unit	Наименование единицы размерности
		CounthStory	Наименование количества товаров на складе
		ExecutionTime	Наименование времени изготовления
		Volume	Наименование объема партии
		Assemblyld	Наименование Id изделия
		Parent_id	Ссылка на родительский компонент

Продолжение таблицы 3.1

Сущность	Описание сущности	Свойства	Описание свойства
Assemlies	Сущность содержащая виды изделий, которые может предоставить фирма клиенту	Id	Уникальный идентификатор
		Name	Наименование вида изделия
		StartComponent_id	
Orders	Сущность, которая хранит заказы на изделие от клиента	Id	Уникальный идентификатор
		Name	Наименование заказа
		WeekNumber	Код недели от 1 до 9
		Count	Наименование количества изделий
		Assembly_id	Наименование Id изделия
MainPlans	Сущность содержащая данные о доходах	Id	Уникальный идентификатор
		Week1	Код вида расчетов изготовления изделия по неделям
		Week2	
		Week3	
		Week4	
		Week5	
		Week6	
		Week7	
		Week8	
		Week9	
		Assembly_id	Наименование Id изделия

В реляционной модели данных отношения связываются с помощью пары ключей: первичного ключа родительского отношения (PrimaryKey - PK) и внешнего ключа дочернего отношения (ForeignKey - FK). Связь по ключевым полям между сущностями показаны в таблице 3.2.

Таблица 3.2 – Связь между сущностями

Родительская сущность	Первичный ключ	Дочерняя сущность	Внешний ключ

Users	Id	-	-

Продолжение таблицы 3.2

Родительская сущность	Первичный ключ	Дочерняя сущность	Внешний ключ
Components	Id		
			Parent_id
		Name	
Assemblies	Id		
			StartComponent_id
MainPlans	Id		
			Assembly_id
Orders	Id		
			Assembly_id

Связь между сущностями осуществляется по правилу «один-ко-многим», то есть один экземпляр родительской сущности может быть связан с несколькими экземплярами дочерней сущности. На основании предметной области для работы подсистемы «ИС ПОСТАВКА МРТ» выделена информация о сущностях системы:

Users, Components, Assemblies, MainPlans, Orders

1. Users .
2. Components.
3. Assemblies.
4. MainPlans.
5. Orders.

ER-диаграмма структуры базы данных подсистемы «ИС ПОСТАВКА МРТ» представлена на рисунке 3.6.

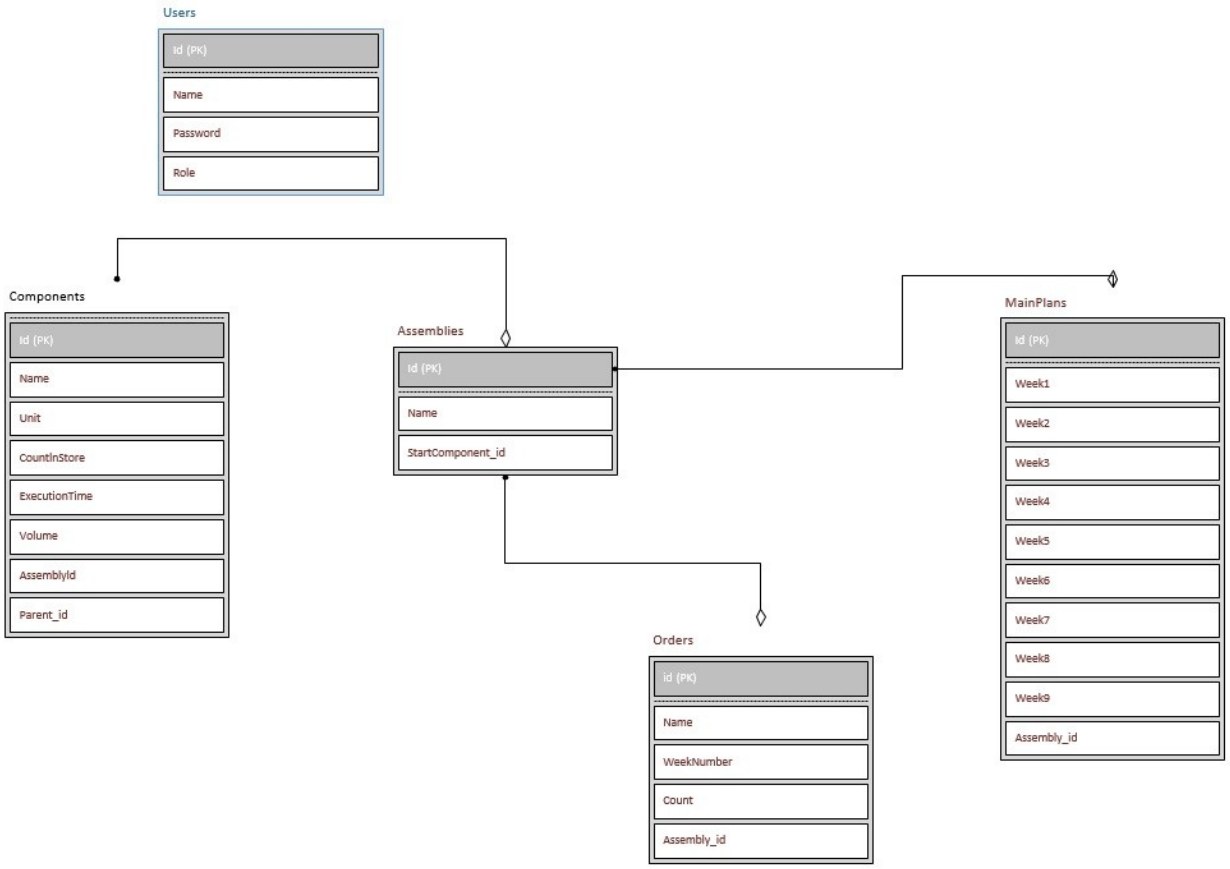


Рисунок 3.6 - ER диаграмма логического уровня

Связь между таблицами определена как Один-ко-Многим, обеспечивается целостность денных, т.е. значения полей в основных таблицах должны содержать поля справочных таблиц [2, 4].

Физическая модель

Физическая модель соответствует уровню даталогического и физического проектирования и позволяет разработать теоретическую логическую и физическую модели БД, соответствующие требованиям и ограничениям выбранной СУБД. Для разработки базы данных используется СУБД MS SQL Server 2017. Структура таблиц базы данных подсистемы физической модели данных представлена в таблицах 3.3 – 3.8.

Таблица 3.3 – Структура таблицы базы данных «Users»

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Id	Int(не NULL)	+	-
Name	nvarchar(max), NULL	-	-
Password	nvarchar(max), NULL	-	-
Role	Int(не NULL)	-	-

Таблица 3.4 – Структура таблицы базы данных «Components»

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Id	Int(не NULL)	+	-
Name	nvarchar(max), NULL	-	-
Unit	nvarchar(max), NULL	-	-
CounthStory	Int(NULL)	-	-
ExecutionTime	Int(NULL)	-	-
Volume	nvarchar(max), NULL	-	-
Assemblyld	Int(NULL)	-	-
Parent_id	Int(NULL)	-	+

Таблица 3.5 – Структура таблицы базы данных «Assemblies»

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Id	Int(не NULL)	+	-
Name	nvarchar(max), NULL	-	-
StartComponent_id	Int(NULL)	-	+

Таблица 3.6 – Структура таблицы базы данных «MainPlans»

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Id	Int(не NULL)	+	-
Week1	Int(не NULL)	-	-
Week2	Int(не NULL)	-	-
Week3	Int(не NULL)	-	-
Week4	Int(не NULL)	-	-

Продолжение таблицы 3.6

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Week5	Int(не NULL)	-	-
Week6	Int(не NULL)	-	-
Week7	Int(не NULL)	-	-
Week8	Int(не NULL)	-	-
Week9	Int(не NULL)	-	-
Assembly_id	Int(NULL)	-	+

Таблица 3.7 – Структура таблицы базы данных «Orders»

Название атрибута	Тип данных	Первичный ключ	Внешний ключ
Id	Int(не NULL)	+	-
Name	nvarchar(max), NULL	-	-
WeekNumber	Int(не NULL)	-	-
Count	Int(не NULL)	-	-
Assembly_id	Int(NULL)	-	+

На физическом уровне производится описание модели данных с учетом требований СУБД MS SQL Server 2017. На рисунке 3.7 представлена физическая модель данных.

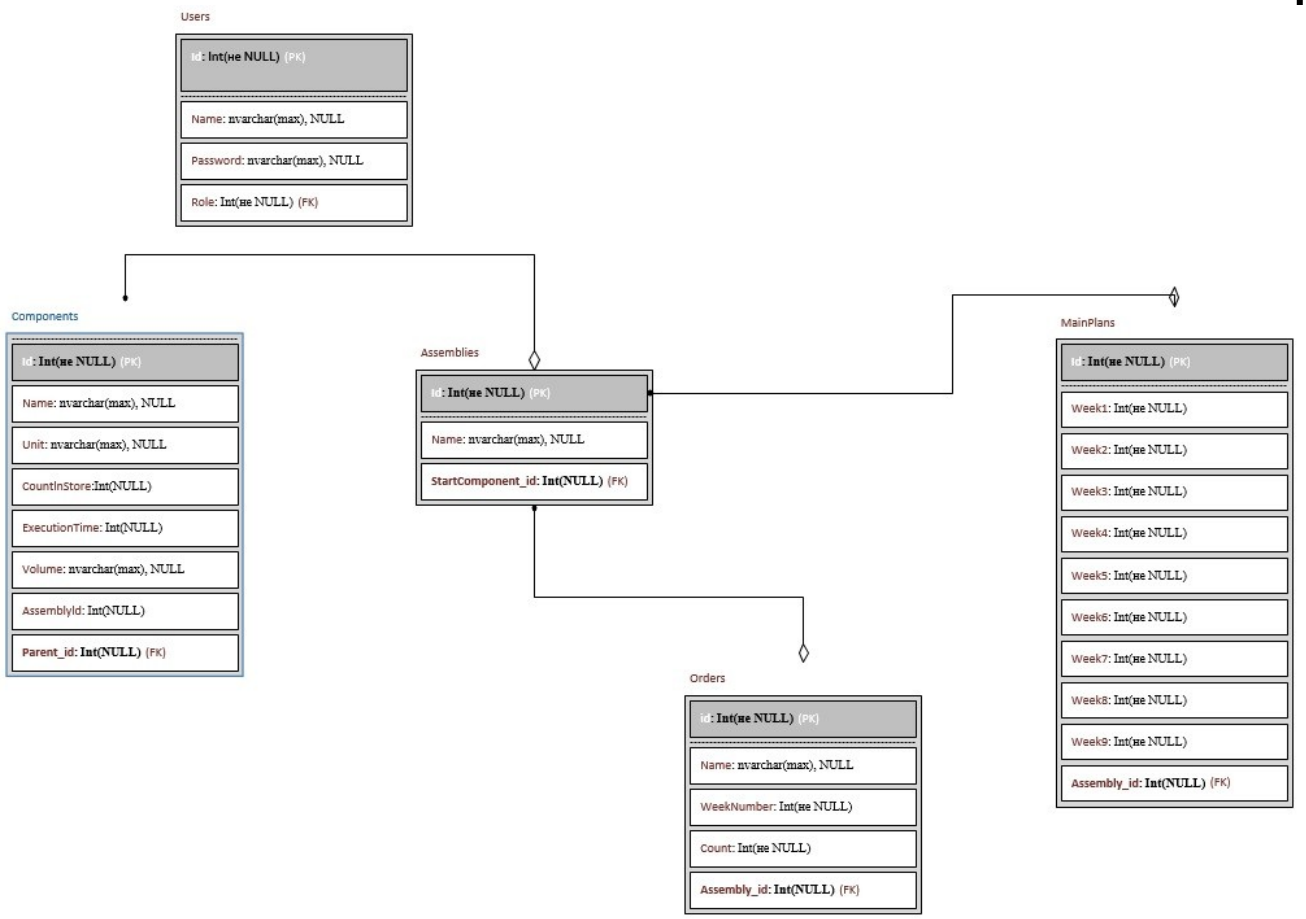


Рисунок 3.7 - Физическая модель данных

Таким образом, была спроектирована физическая модель данных.

3.8. Разработка и описание пользовательского интерфейса

При разработке программы «ИС ПОСТАВКА МТР» были учитаны основные требования пользовательского интерфейса.

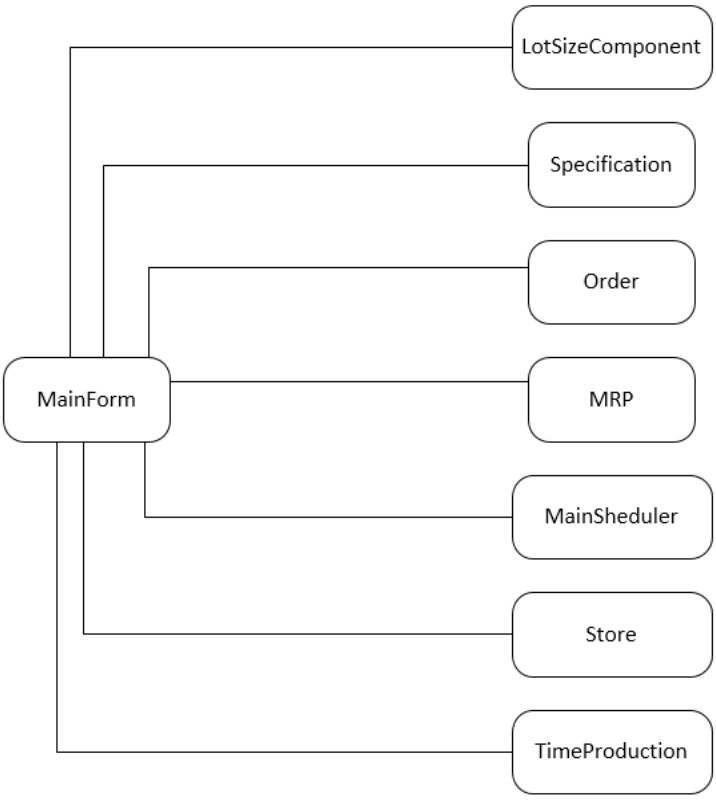


Рисунок 3.6 – Формы программы.

Программа состоит из 8 форм, который выполняют свои задачи.

MainForm – вызывает дочерние формы для частичных задач.

LotSizeComponent – настраивает обьём партии заказа.

Order – занимается оформлением договора на заказ товара

MRP – форма с алгоритмом расчёта и показом планирования по каждой детали.

MainSheduler – форма основного планирования.

Store – форма управляющая товарами на складе.

TimeProduction – форма определяющая время выполнения заказа по каждому компоненту.

Запуск программы осуществляется с прохождения процедуры авторизации.

При запуске программы открывается окно главной формы. (Рис. 3.8)

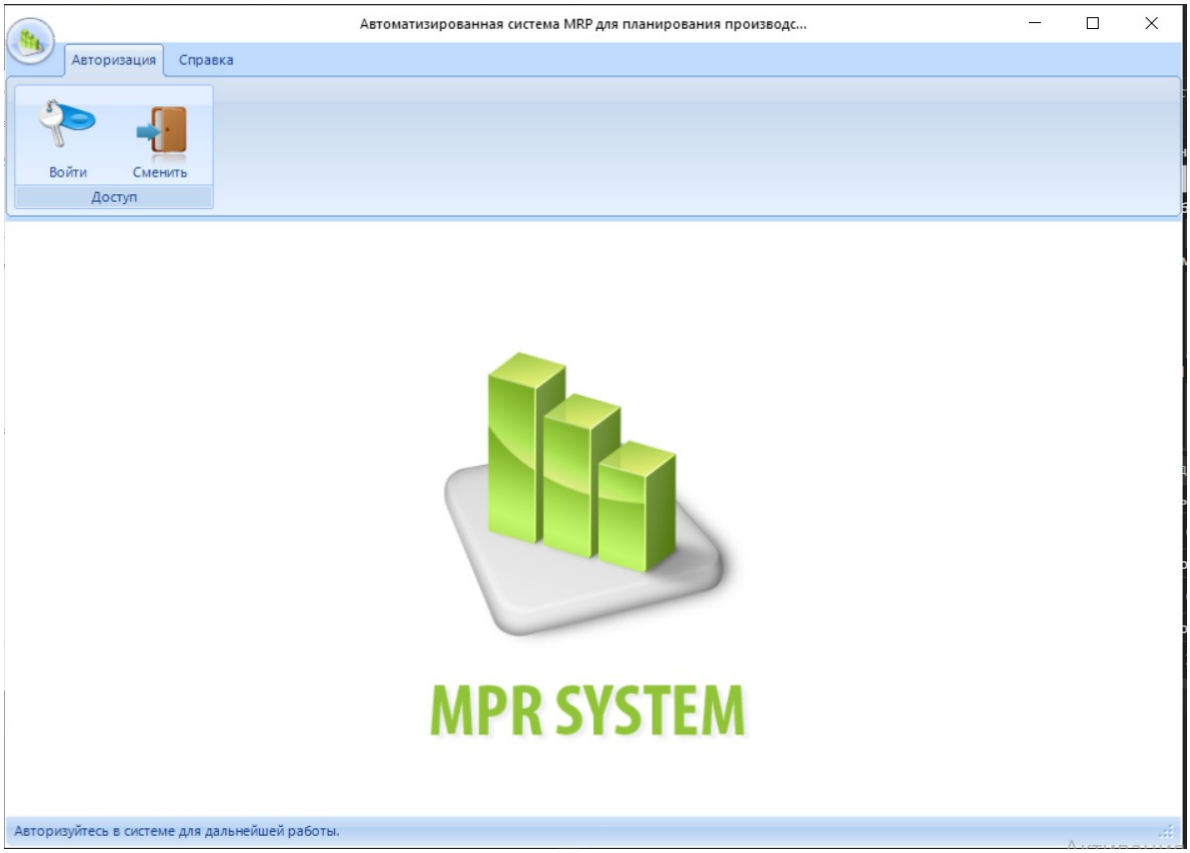


Рисунок 3.8 – Главная форма

В программе три роли: администратор, менеджер и клиент. Для того чтобы зайти в программу необходимо авторизироваться.

При заходе в свою учетную запись открывается основное окно заказа товара (Рис.3.9)

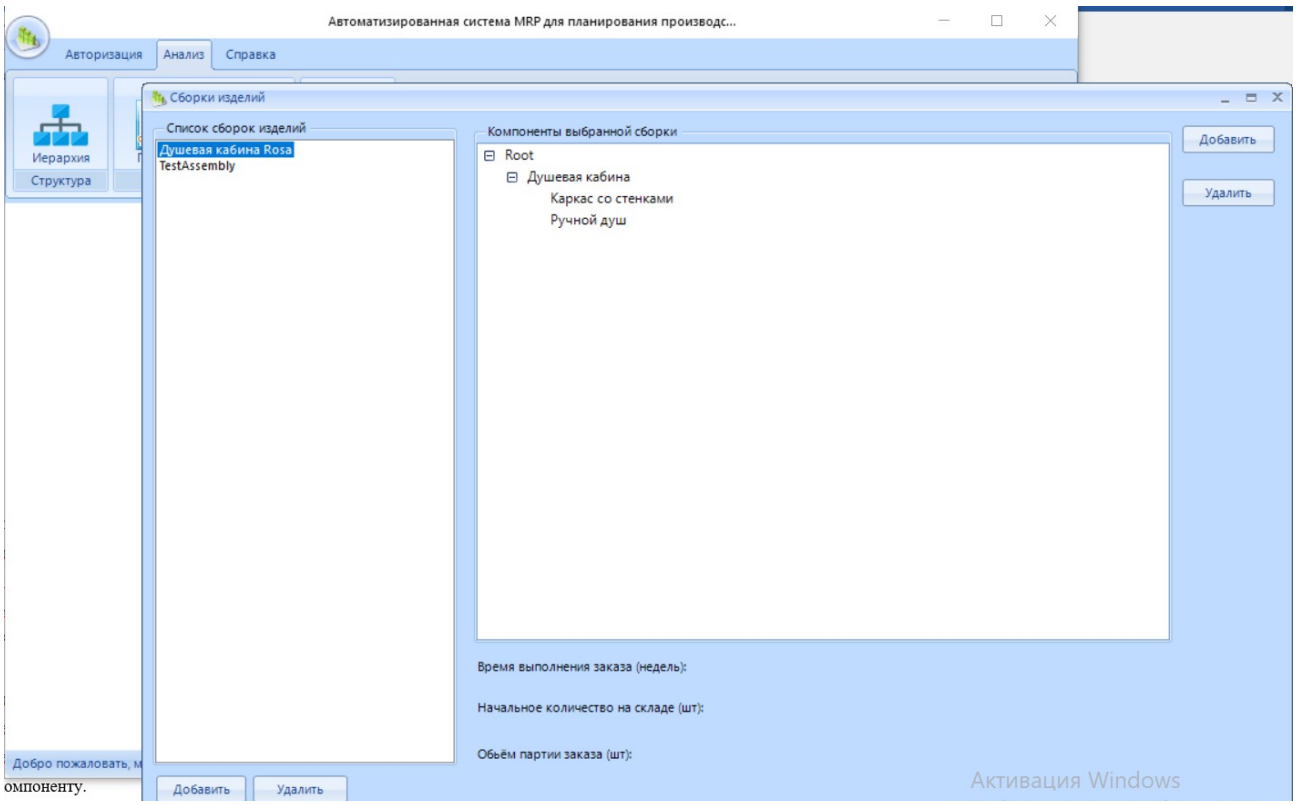


Рисунок 3.9 – Окно заказа и сборки изделий

Выбрав меню «Анализ»-«Планирование компонентов», пользователь может увидеть структуру разрабатываемого изделия, количество на складе необходимых компонентов, и время сборки изделия (Рис. 3.10)

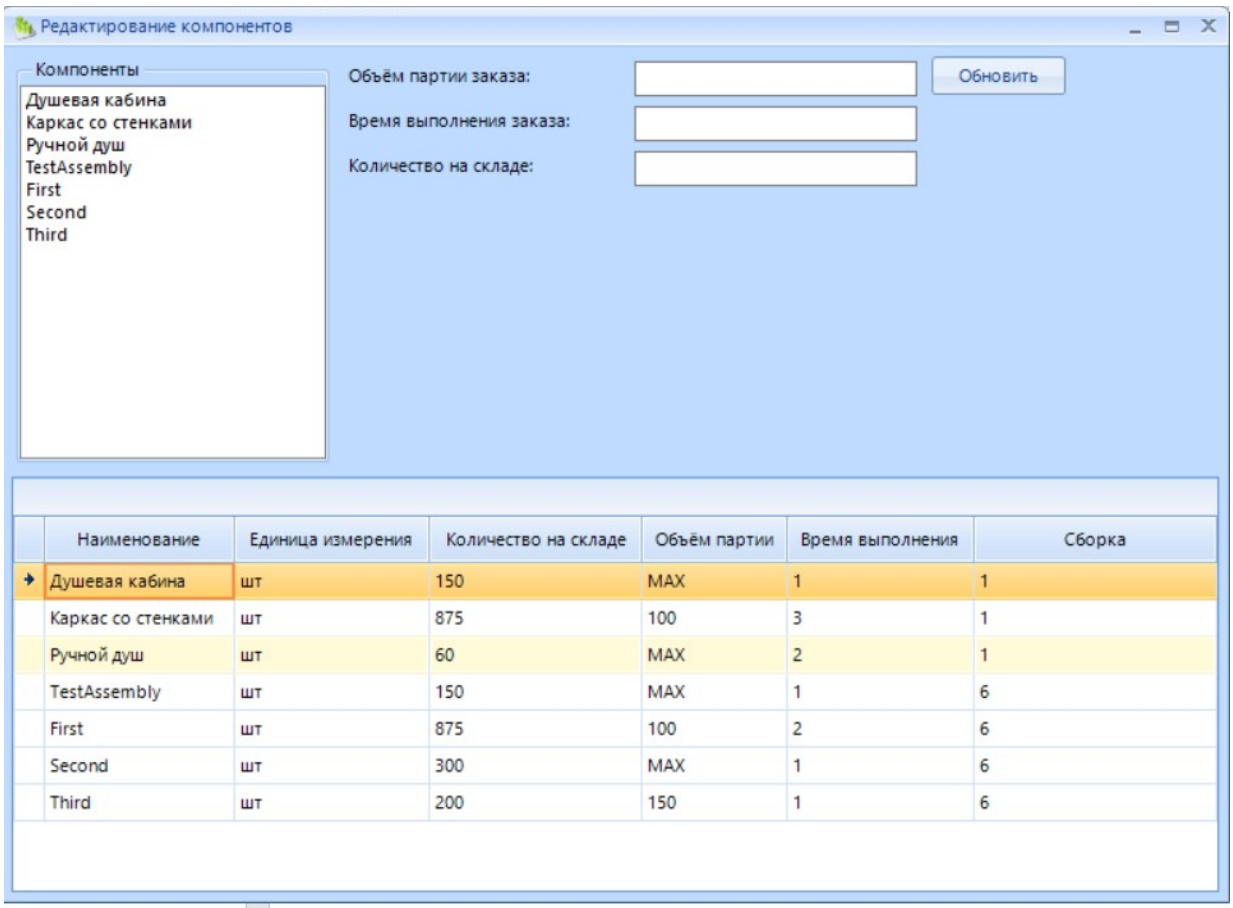


Рисунок 3.10 – Планирование компонентов

Меню «Главное планирование» позволит пользователю увидеть сколько времени будет изготавливаться изделие в целом.

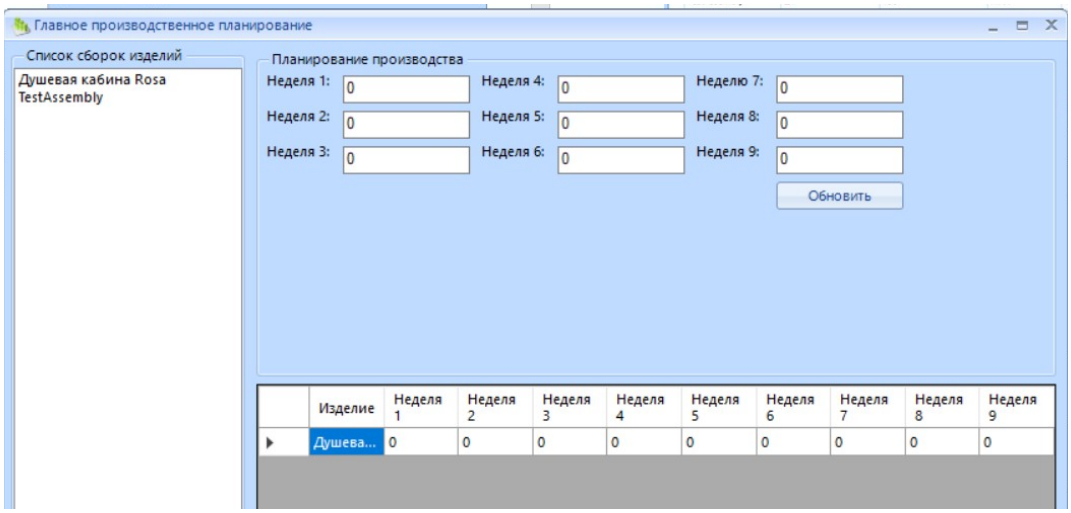


Рисунок 3.11 – Главное производственное планирование

Меню «MRP расчет» позволит пользователю просмотреть и задать анализ алгоритма планирования по конкретной детали изделия (Рис. 3.12)

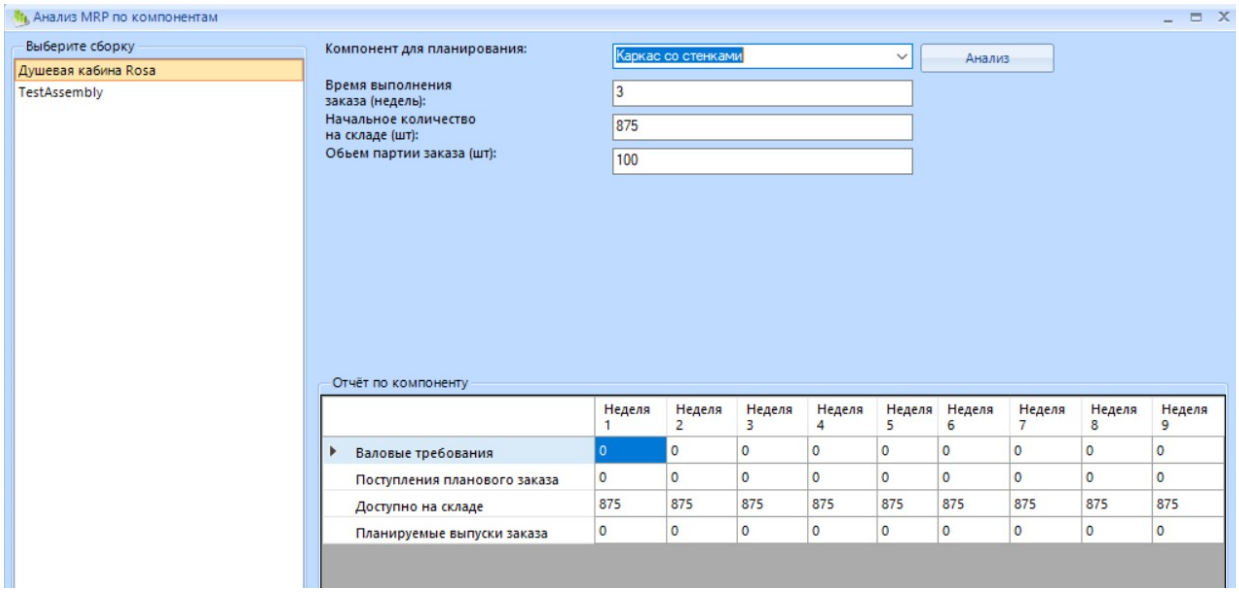


Рисунок 3.12 – MRP рассчет

Только администратор может добавить нового клиента в систему и задать ему определенную роль. Клиент сам не может зарегистрироваться, потому что программа рассчитана для предприятия, а не для общего использования (Рис. 3.13)

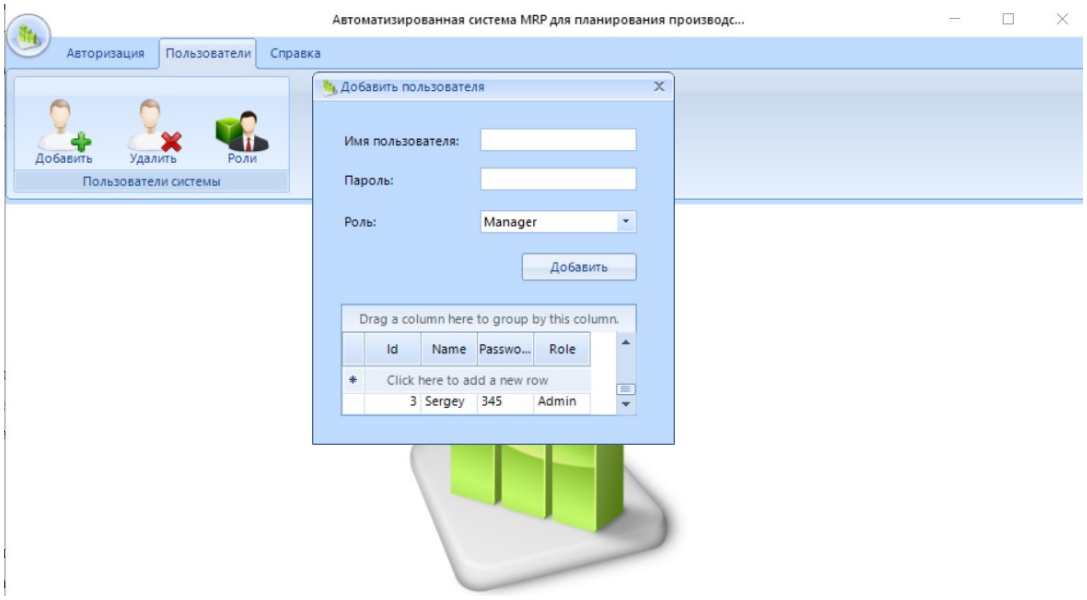


Рисунок 3.13 – Интерфейс «Администратор»

Лингвистическое обеспечение – совокупность средств и правил для формализации естественного языка.[5]

Для построения инфологической модели предметной области использовалось средство разработки структуры БД – Microsoft SQL Server 2017. С помощью наглядной модели БД можно оптимизировать ее структуру и добиться полного соответствия требованиям и задачам организации.

При создании автоматизированной системы использовался язык программирования Visual C#, входящий в состав пакета Microsoft Visual Studio 2017.

При создании базы данных в качестве СУБД использовалась Microsoft SQL Server 2017, использующее язык Transact SQL.

Математическое обеспечение – совокупность математических методов, моделей и алгоритмов, применённых в АС [5].

Алгоритмом называют формально описанную последовательность действий, которые необходимо выполнить для получения требуемого результата.

Алгоритм функционирования автоматизированной системы – алгоритм, задающий условия и последовательность действий компонентов автоматизированной системы при выполнении ею своих функций [5].

Работа ИС «ПОСТАВКА МТР» осуществляется по алгоритму, который представлен на рис. 3.14.

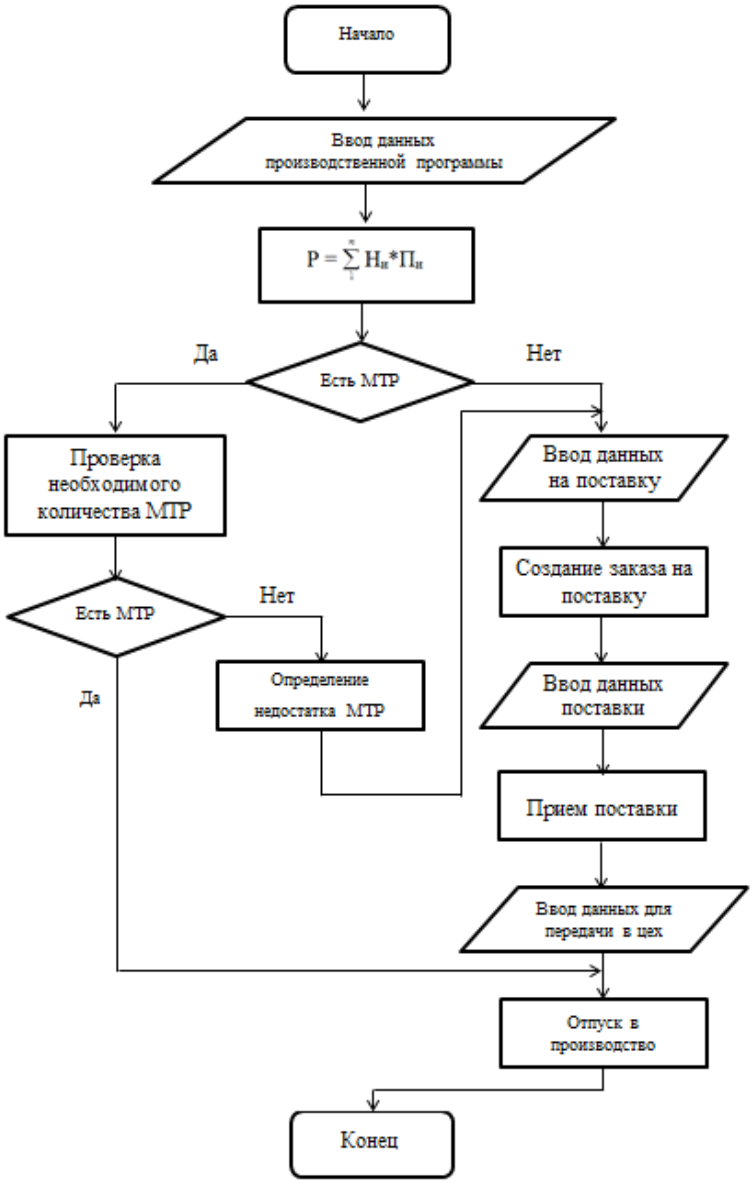


Рисунок 3.14 - Алгоритм работы АС «Поставка МТР»

На основе данных производственной программы определяется потребность в МТР ресурсах для выполнения плана по выпуску продукции.

3.9 Разработка инструкции по эксплуатации ИС

Для лучшего понимая того, как пользоваться разрабатываемой системой, необходимо написать инструкцию по эксплуатации. Такая инструкция поможет пользователю правильно эксплуатировать систему или решить вопросы, связанные с работой отдельных функций. Разработанная инструкция представлена в приложении D.

3.10. Структура ИС

Программа представляет собой централизованную систему сбора, хранения и отображения данных о текущем положении торговых операций. Система предоставляет данные в режиме реального времени.

Программа состоит из следующих комплексов задач и подсистем:

1) комплекс задач по сбору и предоставлению данных о товарах;
2) комплекс задач по загрузке данных из базы данных
3) подсистема формирования отчетности по изготовлению деталей для пополнения склада
4) подсистема формирования отчетности предназначена для выполнения задач формирования отчетов по данным торговых операций и текущим сделкам;
5) подсистема архивирования данных.

3.11. Расчёт производственной программы

Производственная программа состоит из двух разделов: - план производства продукции в натуральном (условно-натуральном) выражении; - план производства в стоимостном выражении. В план производства продукции включаются показатели выпуска изделий определённого ассортимента и номенклатуры в натуральном выражении. При натуральном выражении для индикации объёмов производства применяют такие показатели как: тонны, штуки изделий, кубические и погонные метры, в особенности, на некоторых предприятиях используются двойные измерители. К примеру, планирование производства бумаги закладывается в тоннах и квадратных метрах; в то время как, синтетического волокна - в погонных и квадратных метрах. Использование такого пути в плане даёт более точную характеристику потребительских свойств продукции и физического объема производства. При планировании типов продукции, имеющих разные потребительские качества, но идентичные по назначению применяются условно-натуральные единицы измерения. В качестве иллюстрации, планирование

добычи видов топлива, разных по калорийности, производится в тоннах условного топлива (т.е. за одну тонну принимается масса, эквивалентная 7000 килокалорий). Другие примеры, производство стеновых материалов рассчитывается в миллионах штук условного кирпича, а изготовление консервов - в условных банках. В виде исключения, условные и условно-натуральные измерители также применяются в случае, если предприятия различаются мощностью, габаритами трудоемкостью, но при этом производят один и тот же вид продукции. Следует сказать, что при производстве паровых котлов изделия измеряются в штуках, тысячах тоннах пара в час и квадратных метрах поверхности нагрева, в то время как на турбиностроительных заводах продукция определяется киловаттами мощности турбин, а процесс выпуска шин характеризуются в километрах пробега и штуках (для грузовых автомобилей - в грузоподъемности и штуках). Планирование производства продукции в натуральном выражении позволяет связать выпуск конкретных видов продукции с потребностями рынка, а также с производственными мощностями предприятия и потребностью в ресурсах, которые необходимы для ее производства. Однако для расчета издержки, дохода и прибыли предприятия от реализации продукции, определения общего объема и структуры производства на многопрофильных предприятиях, требуется разработать план производства продукции в стоимостном выражении. План производства продукции в стоимостном выражении содержит расшифровку: 1) реализованной продукции (РП); 2) товарной продукции; 3) валовой продукции. Основным стоимостным показателем этой части плана является реализованная продукция (валовый доход), то есть продукция, оплаченная покупателем или сбытовой организацией. Выраженная стоимость, предназначенных к доставке готовых изделий и полуфабрикатов собственного производства, запасных частей всех видов и назначений, товаров народного потребления, выполняемых работ и оказываемых услуг, реализуемых в соответствии с хозяйственными договорами, потребителями продукции или через собственную сбытовую сеть и является объемом реализованной продукции или валовым доходом. Объем реализованной

продукции ($РП_{пл}$) в стоимостном выражении в соответствии с планом рассчитывают по формуле:

$$РП_{пл} = \sum_{i=1}^{n}(ТП_i \pm ΔОГП_i \pm ОГП_ш)$$

(1)

Где $ТП_i$ - - товарная продукция i-го вида в оптовых ценах предприятия;

$ΔОГП_i$ - - изменение остатков готовой продукции i-го вида на складе предприятия на начало и конец планируемого периода;

$ΔОГП_i$ - - изменение остатков готовой продукции i-го вида, отгруженной, но не оплаченной потребителем на начало и конец планируемого периода; n - количество видов товарной продукции (i=1,2,3,…,n). Реализованная продукция характеризует валовый доход предприятия ($ВД_{пл}$) в плановом периоде. Он определяется как произведение цены на количество проданных единиц изделий или услуг

$$ВД_{пл} = РП = \sum_{i=1}^{n} ВД_i,$$

(2)

$$ВД_i = Ц_i * ОП_{i\ пл},$$

(3)

где $ВД_i$ - валовой доход предприятия от реализации i-го вида продукции; Цi - цена единицы *i*-го вида продукции (оптовая, договорная); ОПi $_{пл}$ - планируемый объем продаж i-го вида продукции (i=1,2,3…,n) в натуральном выражении. Товарная продукция (ТП) выражается в оптовых ценах для увязки плана производства с финансовым планом, а также для определения динамики измерения структуры производства. В понятие стоимости товарной продукции включают стоимости, отображенные на рис. 3.15



Рисунок 3.15 - Стоимость товарной продукции

Плановый объем товарной продукции ($ТП_{пл}$) можно рассчитать по формуле

$$ТП_{пл} = \sum(ПВ_i * Ц_i) + \sum_{j=1}^{m} У_i, \quad (4)$$

где $ПВ_i$ - план выпуска i-го вида продукции в натуральном выражении;

Цi - действующая оптовая цена i-го вида продукции;

n - количество видов товарной продукции (i=1,2,3,…n);

$У_i$ - объем услуг и работ i-го вида промышленного характера;

m - количество видов работ промышленного характера (j=1,2,3,…m).

Валовая продукция включает в себя стоимость всей произведенной продукции и выполненных работ, в том числе незавершенного производства. Ее обычно оценивают в сопоставимых ценах.

Объем валовой продукции (ВП) рассчитывается по формуле:

$$ВП = ТП - ОНЗП_н + ОНЗП_к + ОИН_н + ОИН_к ,$$

(5)

где $ОНЗП_н$, $ОНЗП_к$ - остатки незавершенного производства в стоимостном выражении на конец и начало планового периода соответственно; $ОИН_н$, $ОИН_к$ - остатки инструментов и приспособлении собственного производства на конец и начало планового периода. Изменение остатков незавершенного производства в составе валовой продукции планируется в случае значительного роста объема производства в планируемом периоде в сравнении с отчетным, снятии с производства отдельных видов продукции, создании задела при переходе на выпуск новых изделий со значительной длительностью производственного цикла. По своему составу валовая продукция совпадает с товарной на предприятиях, у которых нет внутрихозяйственного оборота и незавершенного производства. Валовая продукция не совпадает с валовым оборотом, характеризующемся объемом стоимости продукции всех подразделений. Показатели товарной и валовой продукции обладают одним общим недостатком, при этом имея обширное использование в планировании хозяйственной деятельности предприятий, - представление искаженного представления о собственном вкладе в итог деятельности самих предприятий. У этого есть объяснение: включение в их состав стоимости материальных затрат, достигающей на некоторых предприятиях 80-90 % всех издержек. В этом случае более реальная картина о масштабах производства дается посредством показателей чистой и условно чистой продукции. Чистой продукцией характеризуются: заново образованная на предприятии стоимость, в которую не входят издержки на приобретение сырья, материалов, топлива, энергии, а также амортизационные отчисления, которые включены в себестоимость продукции.

Плановый объем стоимости чистой продукции, в которую включаются расходы на оплату труда с начислениями на заработную плату и прибыль предприятия, рассчитывается по формулам:

$$ЧП = РП - МЗ - А,$$

(6)

$$ЧП = ЗП + П_р,$$

(7)

где МЗ - материальные затраты, включаемые в себестоимость продукции;

А - амортизационные отчисления на полное восстановление основных фондов;

ЗП - заработная плата с начислениями на нее;

$П_р$ - прибыль от реализации продукции.

В условно чистую продукцию (УЧП), по сравнению с чистой, включается амортизация и рассчитывается по формуле:

$$УЧП = П_р + МЗ,$$

(8)

$$УЧП = ЗП + А + П_р,$$

(9)

Расчет показателей чистой и условно чистой продукции ведется для исследования структуры производственной программы.

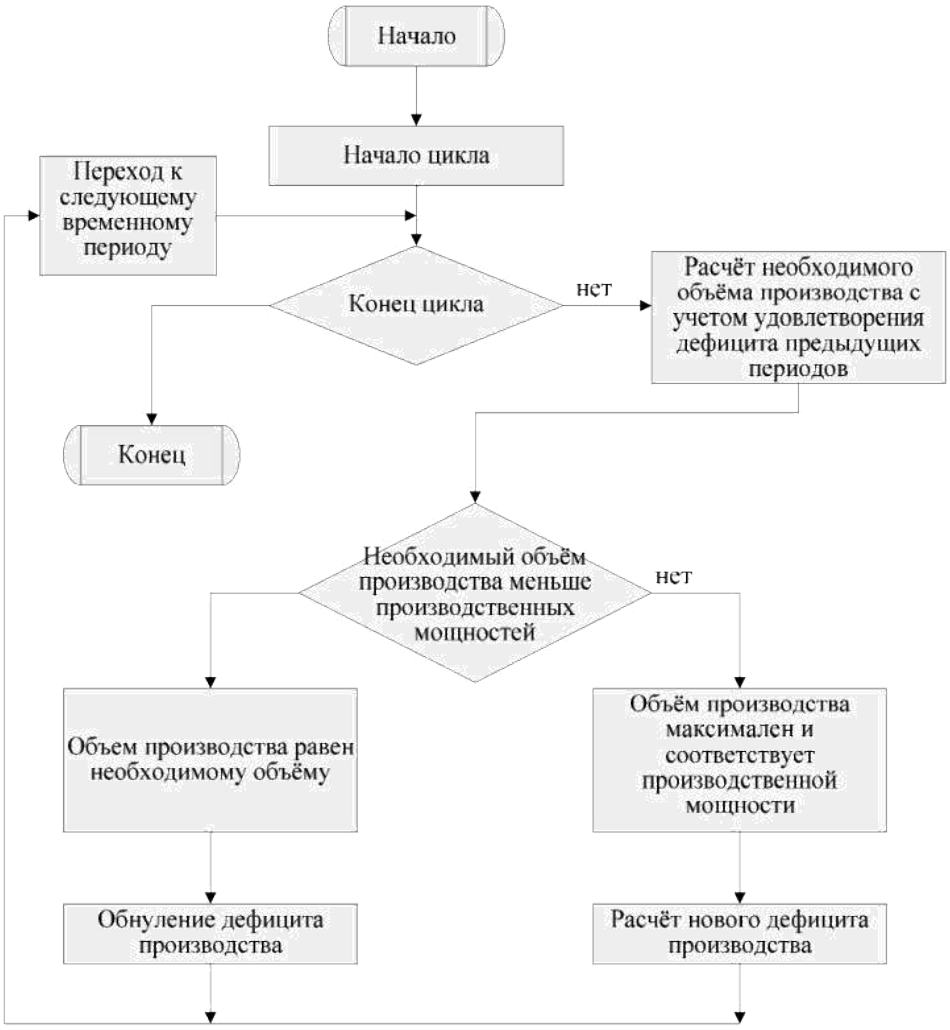


Рисунок 3.16 – Процедура расчёта производственной программы

На рис. 3.16 представлена схема реализации процедуры расчета производственной программы.

В процессе расчета производственной программы цикл по всем периодам планирования операционной деятельности производственного предприятия осуществляется не с первого, а с последнего временного периода. Благодаря этому появляется возможность покрытия нехватки готовой продукции за счёт производства на склад на начальном периоде.

На каждом шаге цикла производится расчёт необходимого объёма производства. При этом происходит определение необходимого объёма из объема производства в целях удовлетворения текущего спроса потребителей объема производства на склад. В случае превышения максимальной производственной мощности рассчитанным объемом необходимого производства предприятия, имеет место планирование максимального выпуска продукции на данный период. Разница между необходимым и минимальным объемами пересчитывается и переносится на следующие периоды. В обратном случае, когда вычисленный объем производства менее производственной мощности, на текущий период сумма объема производства планируется достаточной для удовлетворения текущего спроса потребителей и объема производства для покрытия дефицита в последующие периоды. Затем происходит повтор действий.

Планирование производственной программы начинается с работы с «портфелем заказов» потребителей. Информацию по портфелю предоставляет отдел сбыта. Она формируется в большинстве случаях, основываясь на текущем и перспективном спросе на товар, а также анализируя заключенные или планируемые к заключению договора. При анализе перспективного спроса в зависимости от специфики предприятия происходит использование статистических, эвристических и других методов. Обязательными факторами, которые должны учитываться при планировании ответственными лицами отдела продаж сбыта продукции, являются: влияние разнообразных маркетинговых программ, актуальности спроса, поведение конкурентов и другие немаловажные пункты. В случае использования предприятием производимой продукции на собственные нужды, учет внутренних потребностей добавляется к товарному спросу.

Сдерживающим фактором производства для предприятий, у которых в использовании находится единичное сырье и которые имеют продолжительные связи с поставщиками, может являться поставка сырья и материалов. В подобной

ситуации в роли начальных условий берутся мощности контрактов на поставку сырья и материалов, а для целостности ситуации в снабжении сырьем и матер иалами на начало горизонта планирования следует учесть остатки на складах а также в незавершенном производстве.

«На предприятиях, практикующих снабжение по «точке заказа» и величинам нормативных неснижаемых запасов, входная информация должна быть дополнена нормативными остатками по интервалам планирования, которые могут задаваться в натуральной или временной форме. Временной показатель задает динамический характер нормы запаса, которая рассчитывается исходя из фактического расхода сырья или материала за соответствующий интервал времени» [12].

Технологическая нормативно-справочная информация – это основные входные данные, которые необходимы в расчете производственной программы. К данной части могут быть отнесены пооперационные технологические карты, в которых содержится информация о производственных мощностях, данных изготовления продукции, данные о ремонтах и плановых простоях, разнообразная информация, которая касается производственного процесса на предприятии и обусловлена конкретным типом и спецификой производства. Осуществимость использования производительных сил, время хранения продукции и полуфабрикатов, периоды обеспечения материалами также имеют влияние на результат планирования.

Таким образом, общая технология разработки производственной программы может быть представлена в виде системной функциональной модели, представленной в Приложении А.

Глава 4. Пример расчета производственной программы

4.1. Составление производственного плана и общего плана деятельности

На практике логика работы MRPI системы достаточно проста. Следует рассмотреть пример для понимания работы этой системы. Первый этап заключается в составлении плана деятельности предприятия. Для этого нужно определить производственную программу (MPS) в виде выражения: «Будет производиться 40 комодов в неделю». В будущем, при составлении плана деятельности, следует иметь в виду факторы, приведенные на рис. 4.1.

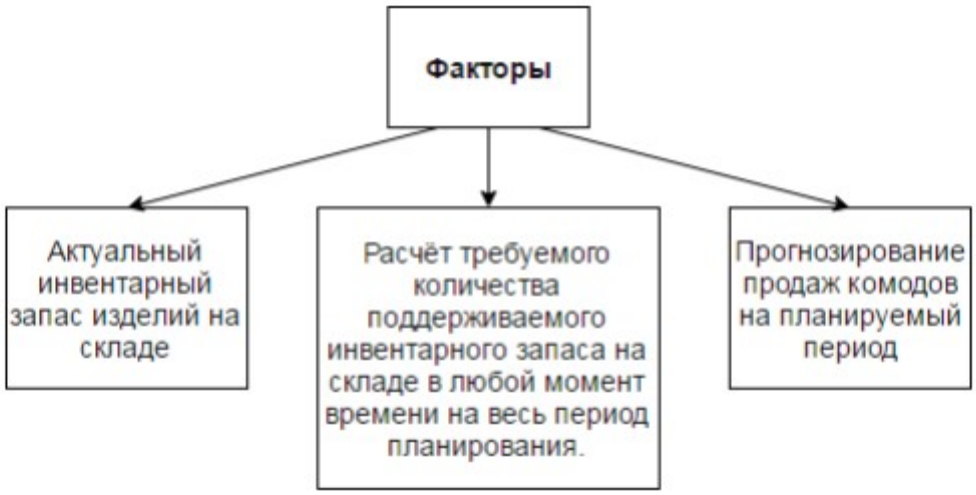


Рисунок 4.1 – Факторы, влияющие на план деятельности

Далее приведена таблица, показывающая типичный план деятельности предприятия:

Таблица 4.1 – План деятельности предприятия

Дата (конец месяца)	План продаж	План	Объем

			производства (MPS)	запасов
31.03	План			70
	Факт			
30.04	План	40	45	75
30.06	Факт	35	46	81
	План	40	45	85
	Факт	42	47	80

В таблице 4.2 отображен типичный бизнес-план, являющийся отображением плана деятельности, с поправкой на финансовый эквивалент.

Таблица 4.2 – Бизнес-план

Дата (конец месяца)		План продаж	План производства (MPS)	Объем запасов
31.03	План			7000
	Факт			
30.04	План	4000	4500	7500
	Факт	3500	4600	8100
31.05	План	4000	4500	8000
	Факт	4800	4200	7500
30.06	План	4000	4500	8500
	Факт	4200	4700	8000
31.12	План	4000	4500	11500
	Факт			

В полный бизнес-план производственного предприятия обязательно включаются затраты на новое производство и совершенствование, а также другие затраты, не связанные с производством и продажами. Относительно системы MRP,

бизнес-план и план деятельности представляются взаимозависимыми, из этого следует, что при обновлении плана деятельности также изменяется бизнес-план. Учитывая главную программу производства как основу, система MRP составляет инвентарный перечень комплектующих-материалов. После этого, система сравнивает его с инвентарными запасами, имеющимися в наличие на складе или в позициях активных заказов, и рассчитывает необходимость в материалах. Приведенный ниже список показывает пример инвентарного списка комплектующих для простого комода.

Таблица 4.3 – Пример инвентарного списка материалов – комплектующих

Инвентарный номер	Название комплектующих	Количество
890997	Крышка	1
789878	Бок правый	1
789887	Бок левый	1
898098	Ручки	10
890988	Направляющие	5
……		
……		
678876	Задняя стенка ящика	5

Подобный инвентарный список обычно называется списком с отступом, в котором элементы списка высшего уровня (комплектующие высшего порядка) располагаются левее, чем их составляющие - комплектующие более низкого порядка. Инвентарные списки являются основой для планирования потребностей в материалах.

4.2. Планирование потребностей в материалах

Следует отметить, что именно модуль планирования потребностей в материалах MRP отмечают как предшественника MRPII-системы. Задача этого

модуля - проектирование поставки всех комплектующих так, чтобы можно было избежать простои производства и уменьшить запасы на складах. В минимизации запаса комплектующих-материалов существует большое количество преимуществ. Например, снижение затрат на хранение, освобождение складских помещений, а также уменьшение замороженных средств, которые были вложены в закуп материала, долгое время ожидающий очереди на конвейер. Входные элементы MRP-модуля показаны на рис. 4.2.

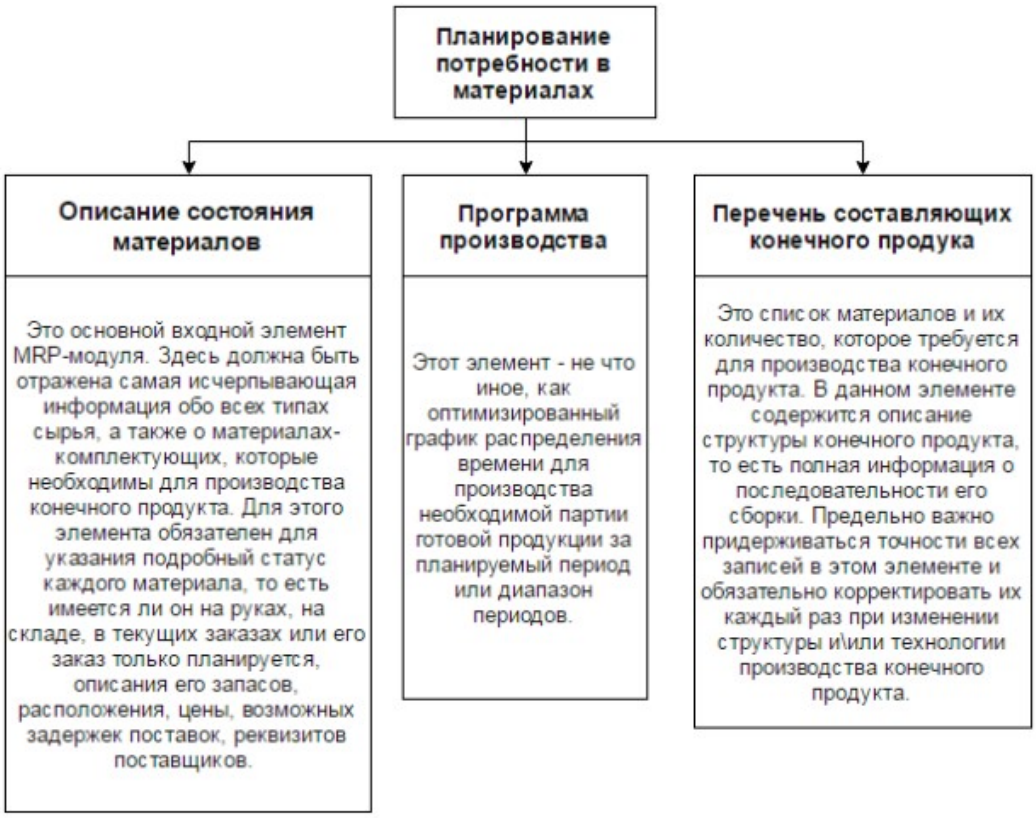


Рисунок 4.2 – Входные элементы MRP-модуля

Принцип работы MRP-модуля заключается в следующем: 1. За каждый период времени на протяжении всего периода планирования и беря за основу инвентарные списки, план производства и текущие запасы на складе создаются данные по полной потребности в материалах. Они показываются в виде интегрированной таблицы, которая выражает потребность в каждом материале, в каждый определенный момент времени.

2. Затем при вычитании из полной потребности тех материалов комплектующих, имеющихся в текущих запасах или занесенных в качестве позиций в активные заказы, определяется чистая потребность. В конечном итоге чистая потребность определяет количество материалов для заказа в каждый определенный момент времени, для удовлетворения текущей потребности производственного процесса. Понятно, что чистая потребность тоже представляет собой определенную таблицу, элементы которой находятся с помощью формулы:

$$\text{Чистая потребность} = \frac{\text{Полная}}{\text{потребность}} _ \frac{\text{Инвентаризировано}}{\text{на руках}} _ \frac{\text{Страховой}}{\text{запас}} _ \frac{\text{Резервирование}}{\text{для других целей}}$$

2. На последнем этапе работы чистая потребность в материалах конвертируется в соответствующий план заказов на требуемые материалы. При необходимости, вносятся поправки в уже действующие планы. Здесь строго учитывается время выполнения каждого заказа, которое, как правило, определяется Поставщиком данного материала. Другими словами MRP система руководствуется известным временем выполнения каждого из заказов (lead time), автоматически составляя план заказов. Данный план заказов является руководящим документом отдела закупок. Соответственно, результаты работы MRP-модуля следующие:

• План Заказов

Помогает рассчитать необходимое количество каждого материала для заказа в любой рассматриваемый промежуток в период срока планирования. План заказов, по сути, является руководителем при дальнейшей работе с поставщиками и определяет производственную программу производства в случае наличия внутреннего производства комплектующих.

• Изменения к плану заказов

4.3. Планирование потребности в производственных мощностях

Производственная программа осуществима, если имеющиеся в наличие производственные мощности могут обработать предписываемое составленным

MRP модулем планом заказов количество сырья и материалов-комплектующих и изготовить из них готовые изделия. Таким образом, MRP-план является основным входным элементом модуля планирования потребностей в производственных мощностях. Другой важный входной элемент модуля - технологическая схема обработки/сборки конечного готового изделия. Эта схема представляет собой определенную таблицу, аналогичную инвентарному списку, только включающую в себя этапы обработки и их длительность, а не комплектующие и их количество. Типичная технологическая схема обработки представлена в таблице 3.4. Прежде всего, для определения производственной мощности предприятия используют производственные единицы. К примеру, такой производственной единицей можно считать инструмент, станок и т.д. Подготовленный CRP-модулем план потребности в производственных мощностях определяет величину стандартных часов работы каждой производственной единицы для обработки нужного объема материала.

Таблица 4.4 – Технологическая схема обработки/сборки готового изделия (начало)

Шаг	Номер производственной единицы	Название работы	Название производственной единицы	Кол-во рабочих часов
1	567787	Расточка	Токарный станок	2
2	67898.456	Шлифовка		6
2.1	67898	Станочн. шлиф.	Шлифовальный станок	5
	456	Ручн. шлиф	Рабочий Михайлов П.И.	2
	….	….	….	…

Очень важно заметить, что модули MRPII-системы однозначно и четко взаимосвязаны. Это означает, что если потребности в материалах не могут быть удовлетворены ни за счет внутреннего производства, ни за счет закупок на стороне,

то в план производства обязательно должны быть внесены изменения. Однако подобные факты должны быть исключениями, т.к. основной задачей является составление успешного производственного плана с самого начала. В таблице 4.5 представлен укороченный вариант типичного плана потребности в производственных мощностях. Этот план готовый выходной элемент CRP-модуля.

Таблица 4.5 – Пример плана потребности в производственных мощностях на примере одной производственной единицы

Номер материала	Номер заказа на пр-во	Кол-во	1.03.16	2.03.16	3.03.16	4.03.16	5.03.16
82345	23489	60		4.5			
91013	0981	600			17.5		
Общее количество часов			394	301	445	310	386

Необходимо подчеркнуть вторично: если в итоге работы CRP-модуля установлена неосуществимость MRP-плана, то требуется пересмотр производственной программы, вдобавок, скорее всего, появится необходимость пересмотра всего плана деятельности. Планировщик, который работает с CRP-системой должен быть достаточно информирован, чтобы знать производственные возможности своего предприятия и понимать задачу компьютера, которая заключается лишь в оптимальности распределения загрузки производственных мощностей на период планирования. Таким образом, планировщик должен попытаться диагностировать и опротестовать очевидно неосуществимый MRP-план до того, как отправить его в CRP-систему, или осуществить поиск пути для расширения производственных мощностей до необходимого уровня.

С момента установления осуществимости плана потребностей в производственных мощностях, берется начало функционирования контроль поддержания установленной производительности. На протяжении всего срока планирования для этого системой систематически происходит создание

контрольных отчетов по производительности. Пример такого отчета приведен в таблице 4.6.

Таблица 4.6 – Пример контрольного отчета по производительности. Установленный период создания отчетов – 1 неделя (начало)

Контрольный отчет для производственной единицы №1500. Дата отчета – 23.05.2016. Пн. Единица измерения – стандартный час работы				
Статус/Дата	2.01.19	9.01.19	16.01.19	23.01.19
План	370	370	370	370
Факт	350	320	390	
Отклонение	-20	-50	-80	

Рассматривая вышеприведенный контрольный отчет, следует обратить внимание на то, что несоответствие реального темпа производства и производственного плана в первую неделю составляло 20 часов, во вторую – 50 и в третью – 80 часов работы. Следовательно, в результате суммарное отклонение достигло 150 стандартных часов.

Условием исправной работы системы является установление величины приемлемого отклонения от плана производства. К примеру, в случае установки того, что величина допускаемого отклонения на начало третьей недели равна половине планового недельного количества часов, то для иллюстрации в таблице 4.6 это отклонение будет равняться 135 часам. То есть во время превышения величины реального отклонения 135 часов, системой будет дан знак о том, что необходимо немедленно вмешаться в работу данной производительной единицы, произвести необходимые мероприятия для повышения ее производительности, вплоть до её возврата на плановый уровень.

Помимо контрольных отчетов производительности, существуют контрольные отчеты потребления материалов-комплектующих для каждой производительной единицы. Такие отчеты имеют место быть для быстрой диагностики ситуаций, в моменты отсутствия развития той или иной производительной единицы в связи с недостаточным снабжением материалами. Контрольный отчет потребления внешне абсолютно похож на отчет, который изображен в таблице 4.6 только вместо связи плановых и реальных часов работы, там отражается расхождение реального и планового потребления материалов рассматриваемой производственной единицей.

Список операций является еще одним необходимым документом, который систематически, а обычно ежедневно, создается MRPII-системой. Обычно формировка списков операций происходит на начало дня, и они передаются мастерам в соответственных производственных цехах. В данных документах имеет место отражение последовательности проведения рабочих операций над сырьем и комплектующими материалами на каждой производственной единице и длительность данного процесса. В таблице 4.7 показан пример списка операций для одной из производственных единиц.

Таблица 4.7 – Пример списка операций

Список операций для производственной единицы №1500 (Токарный станок) на 23.05.16

Номер производственного заказа	Инвентарный номер материала	Количество материала	Дата обработки по плану пр-ва	Кол-во часов обработки
28789	09870	60	20.01.19	3.5
27890	90870	600	23.01.19	19.2

Продолжение таблицы 4.7

Номер производственного заказа	Инвентарный номер материала	Количество материала	Дата обработки по плану	Кол-во часов обработки

			пр-ва	
29895	67418	1200	23.01.19	28.6
78941	89678	600	23.01.19	16.5
58901	98411	220	26.01.19	8.4
Общее количество часов				76.2

Из таблицы понятно, что приведенный список совершает расстановку приоритетов выполнения операций. Задержавшийся по каким-либо причинам производственный заказ от 20.01, MRPII-система поставила первым в очереди. И напротив, у заказа от 26.01.19 имеется минимальный приоритет. Следует заметить, что список операций – не суточный план (это очевидно в связи с тем, что общее количество часов превышает 24), а лишь инструкция для мастера, которая определяет последовательность и содержание производственных операций.

Функции обратной связи также требует внимания. В случае, отсутствия поставки материалов-комплектующих в оговоренные сроки поставщиками, они должны отправить отчет о задержках, сразу, как только получат сообщение о том, что появилась какая-либо проблема. На предприятиях, которые управляются системами класса MRPII, даты поставки учитываются максимально близкими к времени реальной потребности в поставляемых материалах. Следовательно, крайне ответственна заранее постановка системы в известность о вероятных трудностях с заказами, чтобы она могла сгенерировать новый план работы производственных мощностей, опираясь на новый план заказов. В случаях задержки заказов, происходящей достаточно часто, в MRPII-системе задаётся объем минимального поддержания запасов так называемых «ненадежных»

материалов на складе. Главная черта этих систем – это возможность планирования потребности предприятия на небольшие промежутки времени и получать обратную связь, сообщая в системе данные о проблемах в реальном времени. К примеру, при сбоях поставок или поломке оборудования автоматически менять ранее

построенные планы производства. Одной из целей алгоритма работы MRPII-системы является внутреннее моделирование всей области деятельности предприятия. Основная цель алгоритма - учитывать и, используя компьютер, анализировать все внутрекоммерческие и внутрипроизводственные события, происходящие в данный момент и запланированные на будущее. MRPII-система мгновенно среагирует на произошедшее, вне зависимости от того будет ли это допуск брака, или утверждение новых технологических требований, либо произошло изменение программы производства. Система укажет на проблемы, являющиеся результатом произошедших изменений и определит, какие коррективы необходимо внести в производственный план во избежание этих проблем или свод их к минимуму. Конечно, имеет место быть полное устранение последствия того или иного сбоя в производственном, впрочем, MRPII-система посылает информацию о них за максимально длительный промежуток времени и до момента их появления. Так, MRPII-система – надежное средство прогнозирования и оценки последствий внесения тех или иных изменений в производственный цикл, так как она позволяет предвидеть возможные проблемы заранее и осуществлять их предварительный анализ. Любая MRPII-система обладает определенным инструментарием для проведения планирования. Приведенные на рис. 4.3 системные методологии составляют фундаментальные рычаги управления любой MRPII-системы.



Рисунок 4.3 – Рычаги управления

Данные методологии и принципы определяются, исходя из постановки конкретной задачи, применительно к конкретному коммерческому предприятию

4.4. Оценка эффективности внедрения информационной системы на предприятии

Задача выпускной квалификационной работы состоит в исследовании и разработке производственной программы, используя MRP II-систему, позволяющую предприятию при правильном понимании принципов управления MRPII добиться:

• сокращения незавершенного производства, то есть работа не будет выставляться, до времени потребности “точно ко времени” для удовлетворения конечного спроса;

• улучшения обслуживание заказчиков, благодаря своевременному совершению поставок;

• повышения производительности: использование человеческих ресурсов и материалов в соответствии с заказами с минимальными потерями;

• значительного сокращения запасов, позволяющее более экономное использование складских помещений и тем самым уменьшение средств на его хранение;

• создания скоординированной группы управления, позволяющей разрешать оперативные и стратегические вопросы и осуществление организации работы согласно выработанному основному плану производства.

• балансировки запасов: уменьшение дефицита и минимизация запасов, вышедших из использования;

• сокращения цикла производства и цикла выполнения заказа: гибкость реакции бизнеса на спрос;

Расчет полных затрат при выполнении разработки начинается с составления детализированного плана работ, которые следует осуществить для решения

поставленной задачи. Календарный план работы будет основой для разработки детализированного плана. Определение продолжительности каждой работы нужно для расчета затрат на этапе проектирования. В данной работе для определения длительности работ будет использоваться расчетный метод, используя экспертные оценки по формуле:

$$t_j^0 = \frac{3*t_{min}+2*t_{max}}{5},$$

(10)

где t_j^0 – ожидаемая длительность j-й работы;

t_{min} и t_{max} - наименьшая и наибольшая, по мнению эксперта, длительность работы. От продолжительности работ зависит трудоемкость. Результаты расчетов продолжительности работ показаны в таблице 4.8.

Таблица 4.8 – Длительность этапа разработки (начало)

№	Наименование работы	Длительность работы, чел./дни		
		tmin	tmax	t_0
1	Разработка ТЗ и постановка задачи	1	4	2,2
2	Сбор материалов теоретической части	1	2	1,4

3	Исследование видов планирования, производственной программы	7	9	7,8
4	Формирование производственной программы	11	13	11,8
5	Оформление пояснительной записки	1	1	1
6	Оформление иллюстративного материала	1	1	1
Итого:		22	30	25,2

«Для каждого исполнителя необходимо также определить ставку заработной платы за час. Для определения часовой ставки заработной платы необходимо разделить заработную плату (оклад) за месяц на количество рабочих часов в месяце (21 рабочий день * 8 часов = 168 часов)» [13]. Для этого необходимо знать заработную плату исполнителей. Она определена согласно Приказу ректора № 3054 от 26.12.2013 «Об установлении окладов отдельным категориям работников и оптимизации численности централизованных подразделений» для руководителя как заработная плата члена профессорско-преподавательского состава в должности доцента, к.т.н. в размере 19700 руб./мес., для инженера программиста как для инженерно-технического работника вне подразделений в размере 6900 руб./мес. Соответственно, дневная ставка для студента будет равна 6900 руб./мес. разделить на 21 рабочий день, откуда получаем 328,6 руб./день, а для руководителя 938,1 руб./день. Все расчеты сведены в таблицу 4.9.

Таблица 4.9 – Заработная плата (начало)

№	Наименование работы	Исполнитель	Трудоём к., t0, руб./день	Ставка, руб./день
1	Разработка ТЗ и постановка задачи	Инженер	1,2	328,6
		Руководитель	1	938,1
2	Сбор материалов теоретической части	Инженер	1,4	328,6

4	Формирование производственной программы	Инженер	11,8	328,6
5	Оформление пояснительной записки	Инженер	0,8	328,1
		Руководитель	0,2	938,1
6	Оформление работы	Инженер	0,8	328,1
		Руководитель	0,2	938,1
	Итого	Инженер	23,8	
		Руководитель	1,4	

Определение себестоимости продукта

Для оценки эффективности внедрения проекта будет использоваться методология Information Economics. Методология Information Economics ориентирована на объективную оценку портфеля проектов и предусматривает направление ресурсов туда, где они приносят наибольшую выгоду [6]. Идея заключается в том, чтобы заставить информационную службу и бизнес-менеджеров расставить приоритеты и представить более объективные заключения о стратегической ценности отдельных проектов для бизнеса.

Руководителям и бизнес-менеджерам сначала необходимо составить список из главных факторов, влияющих на процесс принятия решения, и оценить относительную значимость («плюсы») и риск («минусы») каждого из них для бизнеса. Для каждого предприятия факторы будут своими, причем они могут добавляться, удаляться или изменяться по мере смены приоритетов. В результате получается полный относительный рейтинг каждого проекта в портфеле информационной службы. Методология Information Economics - быстрый способ определения приоритетов затрат и сопоставления ИТ-проектов с бизнес-целями.

При расчете данным методом были установлены весовые категории (табл. 4.10) для всех критериев, которые будут использоваться для оценивания внедрения системы. Баллы качества по критериям эффективности устанавливались

следующим образом: 0 – функция не выполняется, 1 – функция выполняется частично, 2 – функция выполняется полностью.

Таблица 4.10 - Расчет бальных оценок

№(i)	Критерии эффективности внедрения проектного решения (выполняемые функции)	Балл качества (Bi)		Вес критерия (Pi)
		до	после	
1	Автоматизация планирования ресурсов	0	2	0,65
2	Поддержка процесса оформления заказов	0	2	0,25
3	Уменьшение количества ошибок при ручном вводе данных	0	1	0,1
Σ	Итоговый балл рейтинга качественных показателей эффективности	0	5	1

В результате проведения оценки экономической эффективности, можно сделать вывод о том, что внедрение CRM системы связанной с управлением отношениями клиентами позволяет улучшить существующий процесс статистического анализа показателей, поднять выручку и проходимость клиентов, также дает большие возможности на будущее, например, к масштабированию.

Для более подробного рассмотрения эффекта проекта используем метод оценки экономической эффективности с помощью ЧДД. Будет использоваться расчет показателей общественной эффективности. Основными показателями являются:

- чистый дисконтированный доход (ЧДД);
- индекс доходности (ИД);
- внутренняя норма доходности (ВНД);
- срок окупаемости.

Внедрение CRM системы «TopClean_Sale» является весьма затратным процессом. Но при этом это вложение окажется единовременным и в ближайшие сроки окупится в связи со снижением трудовых и временных затрат (табл.4.11).

Таким образом, в процессе разработки возникают следующие затраты:

Таблица 4.11 – Затраты на разработку аналитического приложения

Статья затрат	Единовременные затраты (руб.)	Текущие затраты (руб.)
Оплата труда: Разработчик (1) Аналитик(1) Администратор(1)	 100 000 70 000	 30 000 в мес.
Амортизация операционной системы, компьютеров, лицензии программного обеспечения и т.д.	-	5 000 в мес.
Покупка программного средства	170 000	
Прочие общехозяйственные затраты		20 000 в мес.
Обучение персонала	10 000	
Накладные расходы		10 000 в мес.
Итого:	390 000	70 000 в мес.

Исходя из анализа процесса внедрения проекта и средних показателей эффективности внедрения АС, была произведена экспертная оценка объема чистой прибыли с учетом внедрения приложения.

Чистый дисконтированный доход (ЧДД) определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу, или как превышение интегральных результатов над интегральными затратами. Величина ЧДД для постоянной нормы дисконта вычисляется по формуле 11:

$$ЧДД = \sum_{t=0}^{T} (R_t - З_t) * \frac{1}{(1+E)^t} \quad (11)$$

- где Rt – результаты, достигаемые на t-ом шаге;
- Зt – затраты, осуществляемые на том же шаге;

- T – горизонт расчета, равный номеру шага расчета, на котором производится ликвидация объекта, равный месяцу;
- $\frac{1}{(1+E)^t}$ – коэффициент дисконтирования;
- E – норма дисконта (в относительных единицах).

Для расчетов возьмем норму дисконта равную 21,5%, которая складывается из ставки рефинансирования 8,25%, инфляции 3,25% и рисков 10%.

Месяцы (T)	1	2	3	4	5	6	7	8	9	10	11	12
Прирост дохода за счет АИС, руб. (Rt)						550 000	600 000	600 000	640 000	670 000	760 000	800 000
Затраты единовременные, руб. (3t1)	270 000	70 000			10 000							
Затраты текущие, руб. (3t2)	40 000	40 000	40 000	40 000	40 000	40 000	40 000	40 000	40 000	40 000	40 000	40 000
Норма дисконта (E)	0,215											
коэф.дисконтирования $\frac{1}{(1+E)^t}$	0,82	0,68	0,56	0,46	0,38	0,31	0,26	0,21	0,17	0,14	0,12	0,10
Rt-3t	-310 000	-110 000	-40000	-40000	-50000	510 000	560 000	560000	600000	630000	720000	760 000
Rt-3t* $\frac{1}{(1+E)^t}$	-255 144	-74 514	-22 301	-18 355	-18 884	158 530	143269,611	117917	103984	89 862	84 527	73 434
ЧДД	382 326											
ИД	1,0923586											
ЧДД накопительный	-255 144	-329 658	-351 960	-370 315	-389 199	-230 668	-87 399	30 519	134 502	224 365	308 891	382 326
ВНД	11%											

Рисунок 4.4 - Расчеты ЧДД

На основании полученного результата ЧДД = 382 326 руб., можно сделать вывод, что внедрение проекта является эффективным т.к. ЧДД положителен (рис.4.4).

Индекс доходности строится из тех же элементов, что и ЧДД (11) и представляет собой отношение суммы приведенных эффектов к величине капиталовложений (12):

$$ИД = \frac{1}{K} * \sum_{t=0}^{T} (R_t - 3_t^+) \frac{1}{(1+E)^t} \quad (12)$$

- где Rt – результаты, достигаемые на t-ом шаге;
- 3t – затраты, осуществляемые на том же шаге;
- T – горизонт расчета, равный номеру шага расчета, на котором производится ликвидация объекта, равный месяцу;
- $\frac{1}{(1+E)^t}$ – коэффициент дисконтирования;
- E – норма дисконта (в относительных единицах);

•К-капиталовложения.

На основе проведенных расчетов получен ИД=1,09. Так как индекс доходности является больше 1, то подсчет ЧДД можно считать верным, а проект эффективным.

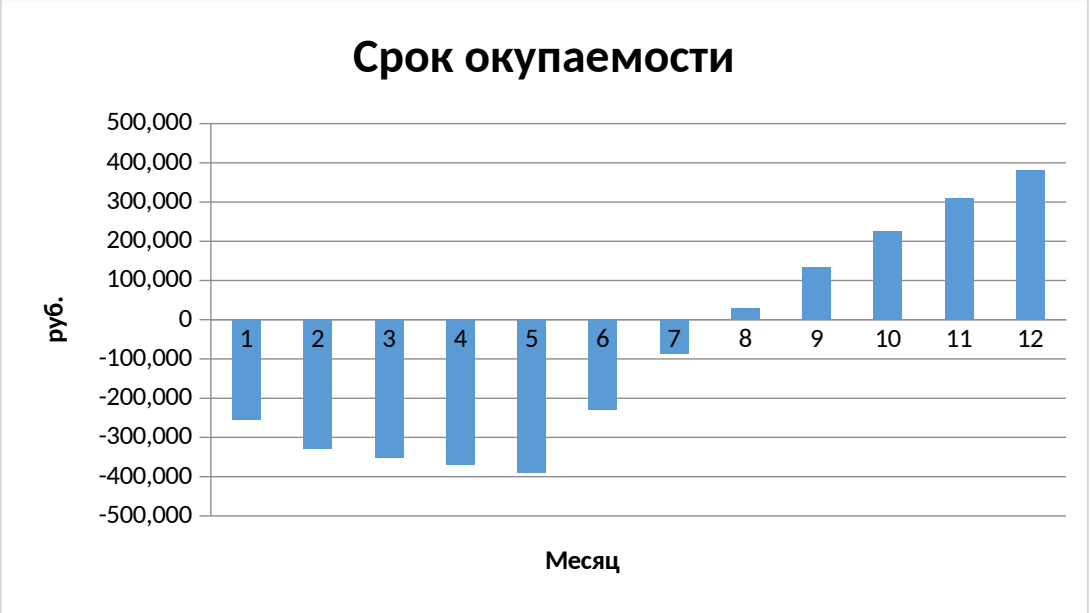


Рисунок 4.5 - Срок окупаемости

Внутренняя норма доходности (ВНД) это максимальный процент, который может быть выплачен для мобилизации капиталовложений в проект. При капиталовложениях равных 350 000 рублей получен ВНД= 11%. Таким образом, при небольших капиталовложениях наблюдается небольшой процент возврата, следовательно, проект оправдывает вложенные деньги.

Срок окупаемости – это период, начиная с которого первоначальные вложения и другие затраты, связанные с проектом, покрываются суммарными результатами его осуществления, что в данном случае достигается на 8 месяце (рис. 4.5). Аналогичные продукты на рынке значительно дороже. Поэтому данная система является экономически выгодной.

Проанализировав подсчеты по экономической эффективности, можно сделать вывод, что данный проект является экономически выгодным с точки зрения реализации.

Заключение

В рамках выполнения данного курсового проекта было выполнено планирование выполнения работы над ВКР путем выявления структуры проекта и составления соответствующего графика, определены и распределены необходимые для проекта ресурсы. Выполнены идентификация и анализ рисков, выделены наиболее значимые из них.

Выполнены разработка части программной реализации информационной системы, описаны основные функции и пользовательский интерфейс.

Совершена работа по описанию экономического обоснования разработки информационной системы контроля качества стерилизации фабрикатов. В результате расчетов было выявлено, что разрабатываемая система экономически выгодна и она позволяет упростить работу и не допустить ошибок в вычислении количества деталей, а далее ее расчёта стоимости.

В результате, в процессе выполнения ВКР были решены следующие задачи:

- рассмотрены разновидности производственных программ
- показаны методы расчета в натуральном и стоимостном выражении различных показателей производственной программы.
- реализация алгоритма MRP I
- создание наилучшего аналога по сравнению с другими программами

Система MRP, взятая для примера, не является идеальной, в ней есть и недостатки, которыми являются неспособность системы планировать человеческие ресурсы; система ориентирована исключительно на заказ; модули системы дают возможность прогнозировать вероятные проблемы, но при этом не делают попыток вовремя решать показанные проблемы, оставляя их на усмотрение сотрудников

предприятия. Но при этом обладает рядом преимуществ: уменьшением запасов; уменьшение затрат на закупку, снижение транспортных затрат по завышенному тарифу, увеличение производительности труда рабочего персонала, простота использования, невысокая цена, снижение затрат на бухгалтерские расчеты.

В заключении следует указать, что поставленные задачи полностью выполнены.

Приложение А
(обязательное)

АВТОРСКАЯ СПРАВКА

Я, _____
автор выпускной квалификационной работы
_____

_____

сообщаю, что мне известно о персональной ответственности автора за разглашение сведений, подлежащих защите законами РФ о защите объектов интеллектуальной собственности.

Одновременно сообщаю, что:

1. При подготовке к защите (опубликованию) выпускной квалификационной работы не использованы источники (документы, отчеты, диссертации, литература и т.п.), имеющие гриф секретности или "Для служебного пользования" ВятГУ или другой организации.

2. Данная работа не связана (связана) с незавершенными исследованиями или уже с завершенными, но еще официально не разрешенными к опубликованию ВятГУ или другими организациями.

3. Данная работа не содержит (содержит) коммерческую информацию, способную нанести ущерб интеллектуальной собственности ВятГУ или другой организации.

1. 4. Данная работа является (не является) результатом НИР или ОКР, выполняемой по договору с организацией (указать согласие заказчика)_____
_____
_____
_____

5. В предлагаемом к опубликованию тексте нет данных по незащищенным объектам интеллектуальной собственности других авторов.

6. Согласен на использование результатов своей работы безвозмездно в ВятГУ для учебного процесса, а также на размещение своей работы в электронной информационно-образовательной среде ВятГУ.

7. Использование моей выпускной квалификационной работы в научных исследованиях оформляется в соответствии с законодательством РФ о защите интеллектуальной собственности.

Автор _____ _____
/личная подпись/ /И. О. Фамилия/

"____"_____ 20_____ г.

2. Сведения по авторской справке подтверждаю:

Заведующий кафедрой _____ _____
/личная подпись/ /И. О. Фамилия/

"____"_____ 20_____ г.

Приложение В
(справочное)

Библиографический список

1. Производственная программа [Электронный ресурс] URL: http://www.ilect.ru/articles/program-production

2. Шеремет А.Д., Старовойтова Е.В. Бухгалтерский учет и анализ: учебник. М.: ИНФРА-М. 2010. 618 с.

3. Кондраков Н.П. Самоучитель по бухгалтерскому учету. 6-е изд., перераб. и доп. М.: Проспект. 2014. 592 с.

4. Бондаренко, А. В. Об алгоритме прогнозирования объема продаж, разработанным Кошечкиным С. А. [Текст] / А. В. Бондаренко // Маркетинг в России и за рубежом. 2012. №5. С. 129 – 141.

5. Царев, В. В. Внутрифирменное планирование [Текст]: учебное пособие / В. В. Царев. Спб.: Питер, 2012. 496 с

6. Электротехника. Бизнес-стратегия 2013 [Электронный ресурс]: Всероссийский деловой Форум. URL: http://www.energoinform.org/news/2013/energy/problemirossiyskoyelektrotekhnicheskoy-promishlennosti.aspx

7. Шатохин, И. В. «Вытягивающее» планирование производства с учетом страхового задела [Электронный ресурс] / Управление производством // И. В. Шатохин, С. Б. Шелкоплясов. URL: http://www.uppro.ru/library/information_systems/production/vityag-planir.htm

8. Туровец О.Г. Организация производства и управление предприятием [Текст] /О.Г. Туровец, В.Б.Родионов, М.И.Бухалков. – М.:ИНФРА–М, 2005. Забодаева И.В. Формирование интегрированной корпоративной информационной системы как инновационное направление совершенствования внутрифирменного планирования и управления // Экономика, статистика и информатика. Вестник УМО. 2010. № 6. С. 52-59.

9. Питеркин С.В., Оладов Н.А., Исаев Д.В., Точно вовремя для России. М.:Альпина Бизнес Букс, 2010. – 368 с.

10. MRP II [Электронный ресурс] URL: http://pl-e.ru/w/MRP_II

11. Толстов И.В. Согласование внутрифирменного планирования на промышленном предприятии /Толстов И.В.// Ползуновский вестник. 2005. № 2. С. 187-192.

12. Дубенецкий В.А., Советов Б.Я., Цехановский В.В., Проектирование корпоративных информационных систем. СПб.: Изд-во СПбГЭТУ "ЛЭТИ", 2013. 191 с.

13. Алексеева О.Г. Методические указания по экономическому обоснованию выпускных квалификационных работ бакалавров: Метод. указания, СПб.: Изд-во СПбГЭТУ “ЛЭТИ”, 2013. с.

Приложение С

(справочное)

Список сокращений и условных обозначений

MRP - Material Requirements Planning (планирование потребности в материалах)

МТР – материально-технические ресурсы

МПЗ – материально-производственные запасы

ERP - Enterprise Resource Planning (планирование ресурсов предприятия)

SaaS - Software as a Service (программное обеспечение как услуга)

МЗ – материальные затраты

СУБД – система управления базами данных

РП – реализованная продукция

ТП – товарная продукция

ОГП – остатки готовой продукции

ВД - валовый доход

ПВ – план выпуска

ОНЗП – остатки незавершенного производства

А – амортизационные отчисления

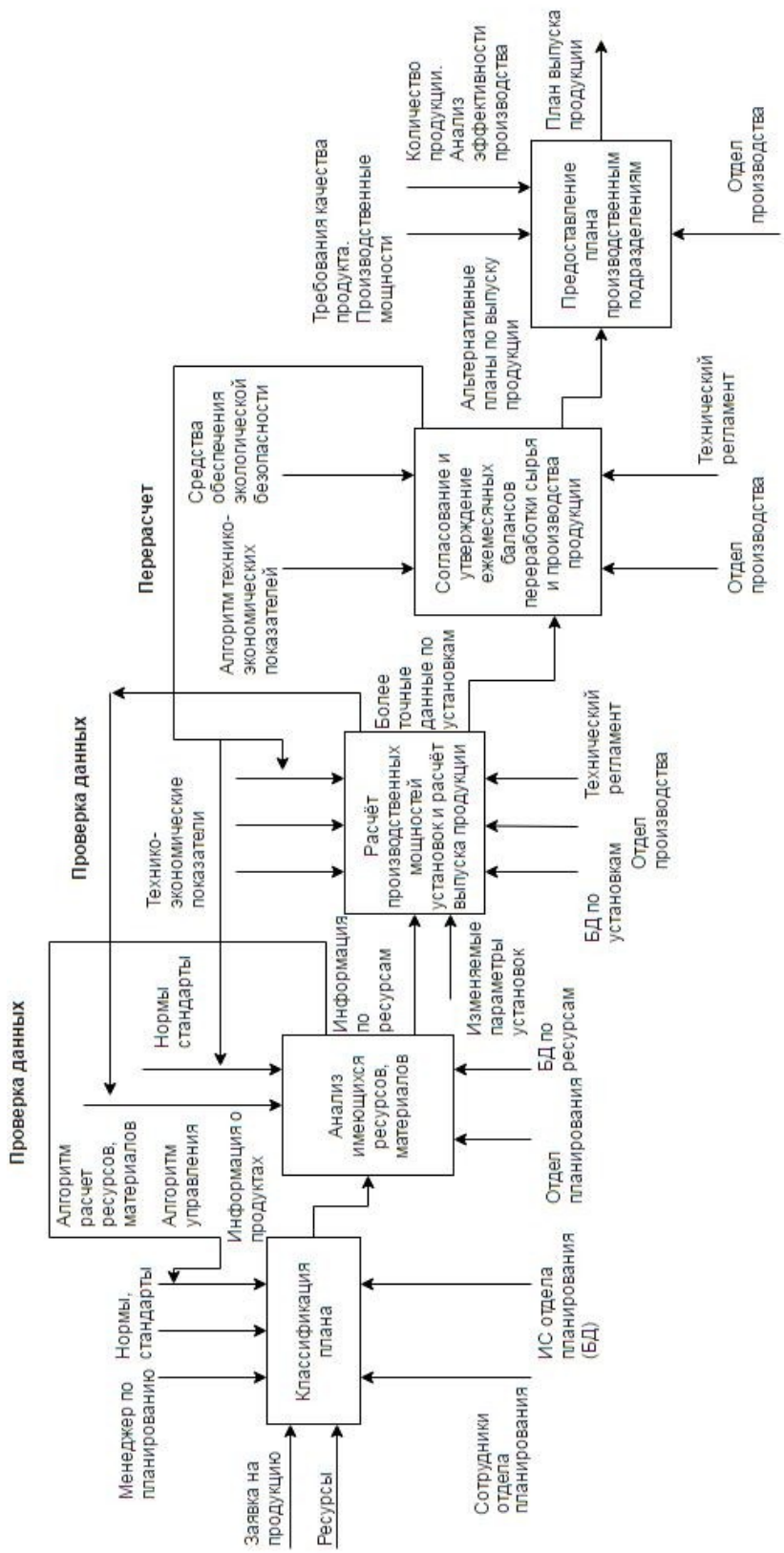
П – прибыль

ЧДД – чистый дисконтированный доход

ВНД – внутренняя норма доходности

Приложение D

(обязательное)



Приложение Е

(справочное)

Инструкция по эксплуатации

Работу в приложении необходимо начать с авторизации пользователя в системе. Регистрировать нового клиента может только Администратор. Он дает клиенту доступ к системе и возможность сделать и отслеживать заказ.

Чтобы заказать изделие клиенту необходимо:

1) зайти под своим логином и паролем;
2) в окне клиенту после успешной авторизации будет доступно меню «Анализ»;
3) в меню «Структура» - «Иерархия» заказывается изделие и выбирается список и количество деталей, которые необходимо изготовить;
4) в меню «Компоненты» клиенту по алгоритму по MRP будет рассчитано время изготовление каждой детали по выбранному товару и сборка изделия в целом, если деталь присутствует на складе.
5) затем в меню «Структура» выбрать детали из которого будет собрано изделие, если деталей нет на складе, то заказ отправляется на производство. Менеджер в своем личном кабинете отправляет заявку на производство и затем оповещает клиента в личном кабинете о времени изготовления заказа.

Чтобы запланировать Менеджеру заказ для этого необходимо:

1) зайти под своим логином и паролем;
2) в окне менеджеру после успешной авторизации будет доступно меню «Заказы»;
3) в выпадающем списке «Заказы» выбрать необходимый заказ, который был прислан системой от клиента
4) распланировать по неделям изготовление деталей заказа.

Для добавления обновленной информации и отправки клиенту новых данных

нажать на кнопку «Добавить заказ»

В блоке «Список заказов» будут выводиться текущие данные, данные в виде таблицы.

Если необходимо добавить нового клиента или присвоить поменять роль пользователя необходимо:

1) зайти под своим логином и паролем;
2) в окне администратору после успешной авторизации будет доступно меню «Пользователи системы»;
3) чтобы добавить нового пользователя нужно в меню «Пользователи системы» - «Добавить» ввести данные нового клиента и присвоить необходимую роль.

В блоке «Список клиентов» будут выводиться текущие данные, данные в виде таблицы.

Приложение F

(справочное)

Программный код

Реализация в среде Visual Studio 2017

Програмный код выложен не полностью, только основные функции.

Функция, отвечающая за авторизацию

Admin

1. Добавление нового пользователя

```
using System;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;

namespace MRP.Admin
{
   public partial class AddUser : Telerik.WinControls.UI.RadForm
   {
      public AddUser()
      {
         InitializeComponent();
         LoadRoles();
         LoadUsers();
      }

      private void LoadRoles()
      {
         radDropDownList1.DataSource = Enum.GetValues(typeof(Role));
         if (radDropDownList1.Items.Count > 0)
         {
            radDropDownList1.SelectedIndex = 0;
         }
      }
```

```
        private void LoadUsers()
        {
            using (var db = new DataContext())
            {
                var users = db.Users.ToList();
                radGridView1.DataSource = users;
            }
        }

        private void radButton1_Click(object sender, EventArgs e)
        {
            if (radTextBox1.Text.Length == 0 || radTextBox2.Text.Length == 0)
            {
                MessageBox.Show(@"Заполните все поля");
            }

            using (var db = new DataContext())
            {

Enum.TryParse<Role>(radDropDownList1.SelectedValue.ToString(), out var
selectedRole);

                var user = new User
                {
                    Name = radTextBox1.Text,
                    Password = radTextBox2.Text,
                    Role = selectedRole
                };
                db.Users.Add(user);
                db.SaveChanges();
                LoadUsers();
            }

        }
    }
}
```

2. Присваивание роли пользователю

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Entity;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MRP.Entities;
using Telerik.WinControls;

namespace MRP.Admin
{
   public partial class ChangeUserRole : Telerik.WinControls.UI.RadForm
   {
      public ChangeUserRole()
      {
         InitializeComponent();
         LoadRoles();
         LoadUsers();
      }


      private void LoadUsers()
      {
         using (var db = new DataContext())
         {
            var users = db.Users.ToList();
            radGridView1.DataSource = users;
         }
```

```
        }

        private void LoadRoles()
        {
            radDropDownList1.DataSource = Enum.GetValues(typeof(Role));
            if (radDropDownList1.Items.Count > 0)
            {
                radDropDownList1.SelectedIndex = 0;
            }
        }

        private void radButton1_Click(object sender, EventArgs e)
        {
            var userId = radGridView1.CurrentRow.Cells[0].Value;
            if (userId == null) MessageBox.Show(@"Выберите пользователя
в таблице");

            using (var db = new DataContext())
            {
                var userToChange = db.Users.FirstOrDefault(x => x.Id ==
(int)userId);
                if (userToChange != null)
                {

Enum.TryParse<Role>(radDropDownList1.SelectedValue.ToString(), out
var selectedRole);
                    userToChange.Role = selectedRole;
                    db.Entry(userToChange).State = EntityState.Modified;
                    db.SaveChanges();
                    LoadUsers();
                }
                else
                {
```

```
                MessageBox.Show(@"Пользователь не найден");
            }
        }
    }
}
```

3. Удаление пользователя из списка

```
using System;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;
using Telerik.WinControls.UI;

namespace MRP.Admin
{
    public partial class DeleteUser : RadForm
    {
        public DeleteUser()
        {
            InitializeComponent();
            LoadUsers();
        }

        private void LoadUsers()
        {
            using (var db = new DataContext())
            {
                var users = db.Users.ToList();
                radGridView1.DataSource = users;
            }
        }

        private void radButton1_Click(object sender, EventArgs e)
        {
            var userId = radGridView1.CurrentRow.Cells[0].Value;
```

```
            if (userId == null) MessageBox.Show(@"Выберите пользователя
    в таблице");

            using (var db = new DataContext())
            {
                var userToDelete = db.Users.FirstOrDefault(x => x.Id == (int)
    userId);
                if (userToDelete != null)
                {
                    db.Users.Remove(userToDelete);
                    db.SaveChanges();
                }
                else
                {
                    MessageBox.Show(@"Пользователь не найден");
                }
            }
        }

    }
}
```

Core

```
using System;
using System.Collections.Generic;
using System.Linq;
using MRP.Entities;
using MRP.Model;

namespace MRP.Core
{
    public static class Utils
    {
```

```
        public static List<Component> GetTreeComponents(Component
parent)
        {
            var components = new List<Component> {parent};
            TraverseTree(parent, components);
            return components;
        }

        private static void TraverseTree(Component parent,
ICollection<Component> components)
        {
            if (parent.Children == null || parent.Children.Count <= 0) return;
            foreach (var child in parent.Children)
            {
                components.Add(child);
                TraverseTree(child, components);
            }
        }

        public static void TraverseEntityComponents(Component component,
Action<Component> action)
        {
            action(component);
            if (component.Children == null) return;
            foreach (var child in component.Children)
TraverseEntityComponents(child, action);
        }

        public static void TraverseComponentTree(ComponentPlanning
component, Action<ComponentPlanning> action)
        {
            action(component);
            if (component.Children == null) return;
```

```
            foreach (var child in component.Children)
        TraverseComponentTree(child, action);
        }

        public static void DeleteComponentTree(int assemblyId, DataContext
        db)
        {
            int componentsCount;
            do
            {
                var components = db.Components.Where(x => x.AssemblyId ==
        assemblyId).ToList();
                componentsCount = components.Count;
                if (componentsCount != 1)
                {
                    var componentsWithoutChildren = components.Where(x =>
        x.Children == null || x.Children.Count == 0);
                    db.Components.RemoveRange(componentsWithoutChildren);
                    db.SaveChanges();
                }
            } while (componentsCount > 1);
        }
    }
}
```

Entities – хранилище для базы данных. Импорт из SQL

1. Хранение данных пользователей

```
namespace MRP.Entities
{
    public class User
    {
        public int Id { get; set; }
```

```
        public string Name { get; set; }

        public string Password { get; set; }

        public Role Role { get; set; }
    }

    public enum Role {
        Manager,
        Client,
        Admin
    }
}
```

2. Хранение заказов на изделие от клиента

```
namespace MRP.Entities
{
    public class Order
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public Assembly Assembly { get; set; }
        public int WeekNumber { get; set; }
        public int Count { get; set; }
    }
}
```

3. Планирование по неделям

```
namespace MRP.Entities
{
    public class MainPlan
    {
        public int Id { get; set; }
```

```
        public Assembly Assembly { get; set; }
        public int Week1 { get; set; }
        public int Week2 { get; set; }
        public int Week3 { get; set; }
        public int Week4 { get; set; }
        public int Week5 { get; set; }
        public int Week6 { get; set; }
        public int Week7 { get; set; }
        public int Week8 { get; set; }
        public int Week9 { get; set; }
    }
}
```

4. Подключение сервера SQL и таблиц

```
using System.Data.Entity;

namespace MRP.Entities
{
    public class DataContext: DbContext
    {
        public DataContext(): base("DbConnection"){ }

        public DbSet<User> Users { get; set; }

        public DbSet<Component> Components { get; set; }

        public DbSet<Assembly> Assemblies { get; set; }

        public DbSet<Order> Orders { get; set; }

        public DbSet<MainPlan> MainPlans { get; set; }

    }
}
```

5. Все добавленные наименования хранятся в этом классе DBCreator, который будет обновляться

```
using System.Collections.Generic;

namespace MRP.Entities
{
   public class DatabaseCreator
   {
      public void CreateUsers()
      {
         using (var db = new DataContext())
         {
                  var user = new User {Name = "Alex", Password = "123", Role =
                                 Role.Manager};
                 var user2 = new User {Name = "Petr", Password = "234", Role =
                                 Role.Client};
               var user3 = new User {Name = "Sergey", Password = "345", Role =
                                 Role.Admin};

            db.Users.Add(user);
            db.Users.Add(user2);
            db.Users.Add(user3);
            db.SaveChanges();
         }
      }

      public void CreateComponents()
      {
         using (var db = new DataContext())
         {
            var washer = new Component
            {
```

```
    Name = "Душевая кабина",
    Unit = "шт",
    Volume = "MAX",
    ExecutionTime = 1,
    CountInStore = 150
};

var firstAssembly = new Assembly
{
    Name = "Душевая кабина Rosa",
    StartComponent = washer
};

db.Assemblies.Add(firstAssembly);
db.SaveChanges();

washer.AssemblyId = firstAssembly.Id;
washer.Children = new List<Component>
{
    new Component {
        Name = "Каркас со стенками",
        Unit = "шт",
        AssemblyId = firstAssembly.Id,
        Volume = "100",
        ExecutionTime = 3 ,
        CountInStore = 875
    },
    new Component
    {
        Name = "Ручной душ",
        Unit = "шт",
        AssemblyId = firstAssembly.Id,
        CountInStore = 55,
```

```
        ExecutionTime = 2,
        Volume = "MAX",
        Children = new List<Component>
          {
            new Component
            {
              Name = "Смеситель",
              Unit = "шт",
              AssemblyId = firstAssembly.Id,
              Volume = "50",
              ExecutionTime =  1,
              CountInStore = 900
            }
          }
    }
};

db.Entry(washer).State = System.Data.Entity.EntityState.Modified;
db.SaveChanges();

var mainPlan = new MainPlan
{
    Assembly = firstAssembly,
    Week1 = 50,
    Week4 = 500,
    Week6 = 500,
    Week7 = 100,
    Week9 = 95
};

db.MainPlans.Add(mainPlan);
db.SaveChanges();
```

```
            List<Order> orders = new List<Order>
            {
                new Order {Assembly = firstAssembly, WeekNumber = 1, Count = 50,
                           Name = "Газпром"},
               new Order {Assembly = firstAssembly, WeekNumber = 4, Count = 500,
                           Name = "Петропром"},
               new Order {Assembly = firstAssembly, WeekNumber = 6, Count = 500,
                           Name = "АчайПром"},
               new Order {Assembly = firstAssembly, WeekNumber = 7, Count = 100,
                           Name = "ЗавКаб"},
                new Order {Assembly = firstAssembly, WeekNumber = 9, Count = 95,
                           Name = "ПитерСталь"}
            };

            db.Orders.AddRange(orders);
            db.SaveChanges();

        }
    }
  }
}
```

6. База данных компонентов

```
using System.Collections.Generic;
// ReSharper disable CommentTypo

namespace MRP.Entities
{
    public class Component
    {
        public int Id { get; set; }
        /// <summary>
        /// Название компонента
        /// </summary>
        public string Name { get; set; }
        /// <summary>
        /// Единица измерения
        /// </summary>
        public string Unit { get; set; }
        /// <summary>
        /// Количество на складе
        /// </summary>
        public int? CountInStore { get; set; }
        /// <summary>
        /// Время выполнения заказа
        /// </summary>
        public int? ExecutionTime { get; set; }
        /// <summary>
        /// Объём партии
        /// </summary>
        public string Volume { get; set; }
        /// <summary>
        /// Сборка
        /// </summary>
```

```
    public int? AssemblyId { get; set; }
    public Component Parent { get; set; }
    public ICollection<Component> Children { get; set; }
 }
```

Реализация основных функций программы

1. Выбор компонентов для изготовления и расчет по неделям

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;
using Telerik.WinControls.UI;

namespace MRP.GUI
{
   public partial class Order : RadForm
   {
      public Order()
      {
         InitializeComponent();
         LoadAssemblies();
         LoadOrders();
      }

      private void LoadOrders()
      {
         using (var db = new DataContext())
         {
            var orders = db.Orders.Include(x => x.Assembly)
               .Select(x => new
               {
```

```
                x.Id,
                x.Name,
                x.WeekNumber,
                x.Count,
                Assembly = x.Assembly.Name
            }).ToList();
        ordersDgv.DataSource = orders;
        ordersDgv.Columns[1].HeaderText = @"Название";
        ordersDgv.Columns[2].HeaderText = @"Неделя";
        ordersDgv.Columns[3].HeaderText = @"Количество";
        ordersDgv.Columns[4].HeaderText = @"Изделие";
    }
}

private void LoadAssemblies()
{
    using (var db = new DataContext())
    {
        var assemblies = db.Assemblies.ToList();
        assembliesDdl.ValueMember = "Id";
        assembliesDdl.DisplayMember = "Name";
        assembliesDdl.DataSource = assemblies;
    }
}

private bool ValidateNewOrderInfo()
{
     return radTextBox1.Text.Length > 0 && assembliesDdl.SelectedValue !=
                null && radTextBox2.Text.Length > 0;
}

private void addOrderBtn_Click(object sender, EventArgs e)
{
```

```
if (!ValidateNewOrderInfo())
{
    MessageBox.Show(@"Проверьте входные данные");
    return;
}


using (var db = new DataContext())
{
            var assembly = db.Assemblies.FirstOrDefault(x => x.Id == (int)
           assembliesDdl.SelectedValue);

    var order = new Entities.Order
    {
        Name = radTextBox1.Text,
        Assembly = assembly,
        Count = int.Parse(radTextBox2.Text),
        WeekNumber = int.Parse(radDropDownList2.Text)
    };

    db.Orders.Add(order);
    db.SaveChanges();

    var plan = GetMainPlan(order.Assembly.Id,db);
    var orders = GetAllAssemblyOrders(order.Assembly.Id,db);

    UpdatePlan(plan, orders);
    db.Entry(plan).State = EntityState.Modified;
    db.SaveChanges();
}

LoadOrders();
}
```

```
private void UpdatePlan(MainPlan plan, List<Entities.Order> orders)
{
        plan.Week1 = orders.Where(x => x.WeekNumber == 1).Sum(x =>
          x.Count);
        plan.Week2 = orders.Where(x => x.WeekNumber == 2).Sum(x =>
          x.Count);
        plan.Week3 = orders.Where(x => x.WeekNumber == 3).Sum(x =>
          x.Count);
        plan.Week4 = orders.Where(x => x.WeekNumber == 4).Sum(x =>
          x.Count);
        plan.Week5 = orders.Where(x => x.WeekNumber == 5).Sum(x =>
          x.Count);
        plan.Week6 = orders.Where(x => x.WeekNumber == 6).Sum(x =>
          x.Count);
        plan.Week7 = orders.Where(x => x.WeekNumber == 7).Sum(x =>
          x.Count);
        plan.Week8 = orders.Where(x => x.WeekNumber == 8).Sum(x =>
          x.Count);
        plan.Week9 = orders.Where(x => x.WeekNumber == 9).Sum(x =>
          x.Count);
}



      private List<Entities.Order> GetAllAssemblyOrders(int assemblyId,
          DataContext db)
{
     return db.Orders.Include(x=>x.Assembly).Where(x=>x.Assembly.Id ==
          assemblyId).ToList();
}

private MainPlan GetMainPlan(int assemblyId,DataContext db)
```

```
{
        return db.MainPlans.Include(x => x.Assembly).FirstOrDefault(x =>
            x.Assembly.Id == assemblyId);
}

private void deleteOrderBtn_Click(object sender, EventArgs e)
{
    if (ordersDgv.SelectedRows.Count == 0)
    {
        MessageBox.Show(@"Сначала выберите строку заказа");
        return;
    }

    var orderId = (int) ordersDgv.SelectedRows[0].Cells[0].Value;

    using (var db = new DataContext())
    {
        var order = db.Orders.Include(x=>x.Assembly).FirstOrDefault(x => x.Id
            == orderId);
        if (order == null)
        {
            MessageBox.Show(@"Заказ не найден");
            return;
        }

        var assemblyId = order.Assembly.Id;

        db.Orders.Remove(order);
        db.SaveChanges();

        var plan = GetMainPlan(assemblyId,db);
        var orders = GetAllAssemblyOrders(assemblyId, db);
        UpdatePlan(plan, orders);
```

```
            db.Entry(plan).State = EntityState.Modified;
            db.SaveChanges();
        }

        LoadOrders();
    }

  }
}
```

2. Реализация алгоритма MRP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;
using MRP.Model;
using Telerik.WinControls.UI;

namespace MRP.GUI
{
    public partial class Mrp : RadForm
    {
        private Manager _manager;
        private List<ComponentReport> _reportComponent;
        private int? _selectedAssemblyId;

        public Mrp()
        {
            InitializeComponent();
            LoadAssemblies();
        }

        private void LoadAssemblies()
```

```
        {
            using (var db = new DataContext())
            {
                var assemblies = db.Assemblies.ToList();
                if (assemblies.Count <= 0) return;

                foreach (var x in assemblies)
                {
                    assemblyList.Items.Add(new RadListDataItem
                    {
                        Text = x.Name,
                        Tag = new SelectedAssembly
                        {
                            AssemblyId = x.Id
                        }
                    });
                }
            }
        }


        private void ClearResults()
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox4.Clear();
            comboBox1.Items.Clear();
            comboBox1.Enabled = false;
            dataGridView1.DataSource = null;
                                                comboBox1.SelectedIndexChanged    -=
    comboBox1_SelectedIndexChanged;
        }
```

```
        private void AnalyzeBtn_Click(object sender, EventArgs e)
        {
            try
            {
                if (!_selectedAssemblyId.HasValue)
                {
                    MessageBox.Show(@"Выберите сборку");
                    return;
                }

                _manager = new Manager();
                 _manager.LoadAssemblyStructureFromDb(_selectedAssemblyId.V
alue);
                _manager.AnalyseData();
                _reportComponent = _manager.MakeReport();

                        MessageBox.Show(@"Вычисления завершены. Выберите
компонент для производства");
                comboBox1.Enabled = true;

                comboBox1.Items.Clear();
                foreach (var report in _reportComponent)
                {
                    comboBox1.Items.Add(report.NameComponent);
                }

                                            comboBox1.SelectedIndexChanged +=
comboBox1_SelectedIndexChanged;
            }
            catch (Exception exception)
            {
                MessageBox.Show(@"Ошибка: "+exception.Message);
            }
```

```
    }

        private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
    {
                                                dataGridView1.DataSource =
_reportComponent[comboBox1.SelectedIndex].Results;
        dataGridView1.Columns[0].HeaderText = @"Неделя 1";
        dataGridView1.Columns[0].Width = 80;
        dataGridView1.Columns[1].HeaderText = @"Неделя 2";
        dataGridView1.Columns[1].Width = 80;
        dataGridView1.Columns[2].HeaderText = @"Неделя 3";
        dataGridView1.Columns[2].Width = 80;
        dataGridView1.Columns[3].HeaderText = @"Неделя 4";
        dataGridView1.Columns[3].Width = 80;
        dataGridView1.Columns[4].HeaderText = @"Неделя 5";
        dataGridView1.Columns[4].Width = 80;
        dataGridView1.Columns[5].HeaderText = @"Неделя 6";
        dataGridView1.Columns[5].Width = 80;
        dataGridView1.Columns[6].HeaderText = @"Неделя 7";
        dataGridView1.Columns[6].Width = 80;
        dataGridView1.Columns[7].HeaderText = @"Неделя 8";
        dataGridView1.Columns[7].Width = 80;
        dataGridView1.Columns[8].HeaderText = @"Неделя 9";
        dataGridView1.Columns[8].Width = 80;


        dataGridView1.Rows[0].HeaderCell.Value = "Валовые требования";
                dataGridView1.Rows[1].HeaderCell.Value = "Поступления
планового заказа";
        dataGridView1.Rows[2].HeaderCell.Value = "Доступно на складе";
```

```
            dataGridView1.Rows[3].HeaderCell.Value = "Планируемые выпуски заказа";
                                dataGridView1.AutoResizeRowHeadersWidth(0, DataGridViewRowHeadersWidthSizeMode.AutoSizeToAllHeaders);

                                                                textBox1.Text = _reportComponent[comboBox1.SelectedIndex].LeadTime;
                                                                textBox2.Text = _reportComponent[comboBox1.SelectedIndex].AvailableBalance;
                                                                textBox4.Text = _reportComponent[comboBox1.SelectedIndex].LotSize;
        }

            private void comboBox1_SelectedIndexChanged_1(object sender, EventArgs e)
        {

        }

            private void assemblyList_SelectedIndexChanged(object sender, Telerik.WinControls.UI.Data.PositionChangedEventArgs e)
        {
            if (assemblyList.SelectedItems.Count <= 0) return;

                                                        var selectedAssembly = (SelectedAssembly)assemblyList.SelectedItems[0].Tag;
            _selectedAssemblyId = selectedAssembly.AssemblyId;

            ClearResults();
        }
    }
}
```

3. Производственный план

```
using System;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;

namespace MRP.GUI
{
   public partial class MainScheduler : Telerik.WinControls.UI.RadForm
   {
      private int _selectedAssemblyId;

      public MainScheduler()
      {
         InitializeComponent();
         LoadAssemblies();
      }

      private void LoadAssemblies()
      {
         using (var db = new DataContext())
         {
            var assemblies = db.Assemblies.ToList();
            if (assemblies.Count <= 0) return;

            foreach (var x in assemblies)
            {
               listView1.Items.Add(new ListViewItem
               {
                  Text = x.Name,
                  Tag = new SelectedAssembly
```

```
                {
                    AssemblyId = x.Id
                }
            });
        }
    }
}

    private void Assembly_SelectedIndexChanged(object sender, EventArgs
e)
    {
        if (listView1.SelectedItems.Count <= 0) return;

                                                var selectedAssembly =
(SelectedAssembly)listView1.SelectedItems[0].Tag;
        _selectedAssemblyId = selectedAssembly.AssemblyId;

        using (var db = new DataContext())
        {
            var plan = db.MainPlans.Include(x=>x.Assembly).FirstOrDefault(x
=> x.Assembly.Id == _selectedAssemblyId);
            if (plan == null) return;

            textBox1.Text = plan.Week1.ToString();
            textBox2.Text = plan.Week2.ToString();
            textBox3.Text = plan.Week3.ToString();
            textBox4.Text = plan.Week4.ToString();
            textBox5.Text = plan.Week5.ToString();
            textBox6.Text = plan.Week6.ToString();
            textBox7.Text = plan.Week7.ToString();
            textBox8.Text = plan.Week8.ToString();
            textBox9.Text = plan.Week9.ToString();
```

```
            dataGridView1.DataSource = ShowPlanTable(plan);
        }
    }

    private DataTable ShowPlanTable(MainPlan plan)
    {
        var planTable = new DataTable("Plan");
        DataColumn[] columns = {
                new DataColumn {Caption = "Assembly", ColumnName =
"Изделие"},
            new DataColumn {Caption = "Week1", ColumnName = "Неделя
1"},
            new DataColumn {Caption = "Week2", ColumnName = "Неделя
2"},
            new DataColumn {Caption = "Week3", ColumnName = "Неделя
3"},
            new DataColumn {Caption = "Week4", ColumnName = "Неделя
4"},
            new DataColumn {Caption = "Week5", ColumnName = "Неделя
5"},
            new DataColumn {Caption = "Week6", ColumnName = "Неделя
6"},
            new DataColumn {Caption = "Week7", ColumnName = "Неделя
7"},
            new DataColumn {Caption = "Week8", ColumnName = "Неделя
8"},
            new DataColumn {Caption = "Week9", ColumnName = "Неделя 9"}
        };
        planTable.Columns.AddRange(columns);
        var items = new object[]
        {
            plan.Assembly.Name,
            plan.Week1,
```

```
            plan.Week2,
            plan.Week3,
            plan.Week4,
            plan.Week5,
            plan.Week6,
            plan.Week7,
            plan.Week8,
            plan.Week9
        };

        var row = planTable.NewRow();
        planTable.Rows.Add(row.ItemArray = items);
        return planTable;
    }

    private void UpdatePlan_Click(object sender, EventArgs e)
    {
        using (var db = new DataContext())
        {
            var plan = db.MainPlans.Include(x=>x.Assembly).FirstOrDefault(x
=> x.Assembly.Id == _selectedAssemblyId);
            if (plan == null) return;

            try
            {
                plan.Week1 = int.Parse(textBox1.Text);
                plan.Week2 = int.Parse(textBox2.Text);
                plan.Week3 = int.Parse(textBox3.Text);
                plan.Week4 = int.Parse(textBox4.Text);
                plan.Week5 = int.Parse(textBox5.Text);
                plan.Week6 = int.Parse(textBox6.Text);
                plan.Week7 = int.Parse(textBox7.Text);
                plan.Week8 = int.Parse(textBox8.Text);
```

```
                plan.Week9 = int.Parse(textBox9.Text);

                db.Entry(plan).State = EntityState.Modified;
                db.SaveChanges();

                dataGridView1.DataSource = ShowPlanTable(plan);
            }
            catch (Exception)
            {
                MessageBox.Show(@"Проверьте входные данные");
            }
        }
    }
}
```

4. Главная форма

```
using System;
using MRP.Admin;
using MRP.Entities;
using Telerik.WinControls;

namespace MRP.GUI
{
    public partial class MainForm : Telerik.WinControls.UI.RadForm
    {
        public MainForm()
        {
            InitializeComponent();
            ribbonTab1.IsSelected = true;
            //var creator = new DatabaseCreator();
```

```
        //creator.CreateUsers();
        //creator.CreateComponents();
        //setPermissionsForManager("RootManager");
    }

    private void radMenuItem1_Click(object sender, EventArgs e)
    {

    }

    private void radMenuItem7_Click(object sender, EventArgs e)
    {

    }

    private void radButtonElement4_Click(object sender, EventArgs e)
    {
        new MainScheduler().Show();
    }

    private void radButtonElement6_Click(object sender, EventArgs e)
    {
    }

    private void radButtonElement7_Click(object sender, EventArgs e)
    {
    }

    private void radButtonElement8_Click(object sender, EventArgs e)
    {
        new EditComponents().Show();
    }
```

```
private void radButtonElement5_Click(object sender, EventArgs e)
{
    new Mrp().Show();
}

private void radMenuItem3_Click(object sender, EventArgs e)
{
    Close();
}

private void radButtonElement10_Click(object sender, EventArgs e)
{
    new Order().Show();
}

private void radButtonElement9_Click(object sender, EventArgs e)
{
    new About().Show();
}

private void radButtonElement14_Click(object sender, EventArgs e)
{
    new EditComponents().Show();
}


private void radButtonElement16_Click(object sender, EventArgs e)
{
    new Mrp().Show();
}

private void radButtonElement12_Click(object sender, EventArgs e)
{
```

```
    new MainScheduler().Show();
}

private void radButtonElement1_Click(object sender, EventArgs e)
{
    var loginForm = new Login();
    loginForm.OnUserAuthorized += setUserPermission;
    loginForm.Show();

}

private void radMenuItem13_Click(object sender, EventArgs e)
{

}


private void setUserPermission(object sender, LoginEventArgs e)
{
    resetUserPermission();

    switch (e.UserRole)
    {
        case Role.Manager:
            setPermissionsForManager(e.UserName);
            break;
        case Role.Client:
            setPermissionsForClient(e.UserName);
            break;

        case Role.Admin:
            setPermissionsForAdmin(e.UserName);
            break;
```

```
            default:
                resetUserPermission();
                break;
        }
    }

    private void setPermissionsForManager(string managerName)
    {
        radLabelElement1.Text = @"Добро пожаловать, менеджер " +
managerName;
        ribbonTab2.Visibility = ElementVisibility.Visible;
        ribbonTab5.Visibility = ElementVisibility.Collapsed;
        ribbonTab6.Visibility = ElementVisibility.Collapsed;
        ribbonTab2.IsSelected = true;
        radMenuItem7.Visibility = ElementVisibility.Visible;
        radMenuItem13.Visibility = ElementVisibility.Collapsed;
        radMenuItem14.Visibility = ElementVisibility.Collapsed;
    }

    private void setPermissionsForClient(string clientName)
    {
        radLabelElement1.Text = @"Добро пожаловать, клиент " +
clientName;
        ribbonTab2.Visibility = ElementVisibility.Collapsed;
        ribbonTab5.Visibility = ElementVisibility.Visible;
        ribbonTab6.Visibility = ElementVisibility.Collapsed;
        ribbonTab5.IsSelected = true;
        radMenuItem7.Visibility = ElementVisibility.Collapsed;
        radMenuItem13.Visibility = ElementVisibility.Visible;
        radMenuItem14.Visibility = ElementVisibility.Collapsed;
    }

    private void setPermissionsForAdmin(string adminName)
```

```
        {
            radLabelElement1.Text = @"Добро пожаловать, администратор " +
adminName;
            ribbonTab2.Visibility = ElementVisibility.Collapsed;
            ribbonTab5.Visibility = ElementVisibility.Collapsed;
            ribbonTab6.Visibility = ElementVisibility.Visible;
            ribbonTab6.IsSelected = true;
            radMenuItem7.Visibility = ElementVisibility.Collapsed;
            radMenuItem13.Visibility = ElementVisibility.Collapsed;
            radMenuItem14.Visibility = ElementVisibility.Visible;
        }

        private void resetUserPermission()
        {
            radLabelElement1.Text = @"Авторизуйтесь в системе для
дальнейшей работы.";
            ribbonTab2.Visibility = ElementVisibility.Collapsed;
            ribbonTab5.Visibility = ElementVisibility.Collapsed;
            ribbonTab6.Visibility = ElementVisibility.Collapsed;
            ribbonTab1.IsSelected = true;
            radMenuItem7.Visibility = ElementVisibility.Collapsed;
            radMenuItem13.Visibility = ElementVisibility.Collapsed;
            radMenuItem14.Visibility = ElementVisibility.Collapsed;
        }

        private void radButtonElement11_Click(object sender, EventArgs e) =>
resetUserPermission();

        private void radButtonElement17_Click(object sender, EventArgs e)
        {
            var addUserForm = new AddUser();
            addUserForm.Show();
        }
```

```
        private void radButtonElement18_Click(object sender, EventArgs e)
        {
            var deleteUserForm = new DeleteUser();
            deleteUserForm.Show();
        }

        private void radButtonElement19_Click(object sender, EventArgs e)
        {
            var changeUserRoleForm = new ChangeUserRole();
            changeUserRoleForm.Show();
        }

        private void radButtonElement3_Click(object sender, EventArgs e)
        {
            var assemblyForm = new AssemblyForm();
            assemblyForm.Show();
        }
    }
}
```

5. Авторизация

```
using System;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;

namespace MRP.GUI
{
    public partial class Login : Telerik.WinControls.UI.RadForm
    {
```

```
        public event EventHandler<LoginEventArgs> OnUserAuthorized ;

        public Login()
        {
            InitializeComponent();
        }

        private void radButton2_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void radButton1_Click(object sender, EventArgs e)
        {
             if (radTextBox1.Text.Length == 0 || radButton2.Text.Length ==
0)
            {
                MessageBox.Show(@"Введите логин и пароль");
            }

            using (var db = new DataContext())
            {
                    var foundUser = db.Users.FirstOrDefault(x => x.Name ==
radTextBox1.Text && x.Password == radTextBox2.Text);
                if (foundUser == null)
                {
                    MessageBox.Show(@"Пользователь не найден");
                    return;
```

```
            }

            var args = new LoginEventArgs
            {
                UserName = foundUser.Name,
                UserRole = foundUser.Role
            };

            OnUserAuthorized?.Invoke(this, args);
            Close();
        }
    }
}

public class LoginEventArgs : EventArgs
{
    public string UserName { get; set; }
    public Role UserRole { get; set; }
}
```

6. Заказ необходимых деталей для изделия

```
using System;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Windows.Forms;
using MRP.Entities;
```

```
using Telerik.WinControls.UI;

namespace MRP.GUI
{
    public partial class EditComponents : RadForm
    {
        private int _selectedComponentId;

        public EditComponents()
        {
            InitializeComponent();
            LoadComponents();
            listView1.HideSelection = false;
            LoadMainTable();
        }

        private void LoadComponents()
        {
            using (var db = new DataContext())
            {
                var allComponents = db.Components.ToList();
                if (allComponents.Count <= 0) return;
                foreach (var x in allComponents)
                    listView1.Items.Add(new ListViewItem
                    {
                        Text = x.Name,
                        Tag = x.Id
```

```
                });
            }
        }

        private void ComponentList_SelectedIndexChanged(object sender,
EventArgs e)
        {
            if (listView1.SelectedItems.Count <= 0) return;

            _selectedComponentId = (int) listView1.SelectedItems[0].Tag;

            using (var db = new DataContext())
            {
                var currentComponent = db.Components.FirstOrDefault(x =>
x.Id == _selectedComponentId);
                if (currentComponent != null)
                {
                    textBox1.Text = currentComponent.Volume;
                                                        textBox2.Text    =
currentComponent.ExecutionTime?.ToString();
                                                        textBox3.Text    =
currentComponent.CountInStore?.ToString();
                }
            }

        }

        private void LoadMainTable()
```

```
        {
            using (var db = new DataContext())
            {
                var allComponents = db.Components.ToList();

                var allComponentsTable = new DataTable("AllComponents");
                DataColumn[] columns =
                {
                        new DataColumn {ColumnName = "Name", Caption =
"Наименование"},
                        new DataColumn {ColumnName = "Unit", Caption =
"Единица измерения"},
                    new DataColumn {ColumnName = "CountInStore", Caption =
"Количество на складе"},
                        new DataColumn {ColumnName = "Volume", Caption =
"Объём партии"},
                    new DataColumn {ColumnName = "TimeExecute", Caption =
"Время выполнения"},
                        new DataColumn {ColumnName = "Assembly", Caption =
"Сборка"}
                };

                allComponentsTable.Columns.AddRange(columns);

                foreach (var component in allComponents)
                {
                    var items = new object[]
                    {
                        component.Name,
```

```
                    component.Unit,
                    component.CountInStore,
                    component.Volume,
                    component.ExecutionTime,
                    component.AssemblyId
                };

                var row = allComponentsTable.NewRow();
                allComponentsTable.Rows.Add(row.ItemArray = items);
            }

            radGridView1.DataSource = allComponentsTable;
            radGridView1.BestFitColumns();
        }
    }

    private void UpdateComponent_Click(object sender, EventArgs e)
    {
        using (var db = new DataContext())
        {
            var currentComponent = db.Components.FirstOrDefault(x =>
x.Id == _selectedComponentId);
            if (currentComponent == null) return;

            currentComponent.Volume = textBox1.Text;

                    if (int.TryParse(textBox2.Text, out var executionTime))
currentComponent.ExecutionTime = executionTime;
```

```
                    if (int.TryParse(textBox3.Text, out var countInStore))
currentComponent.CountInStore = countInStore;

                db.Entry(currentComponent).State = EntityState.Modified;
                db.SaveChanges();
            }

            LoadMainTable();
        }
    }
}
```

8. Производство изделия

```
using System;
using System.Data.Entity;
using System.Linq;
using System.Windows.Forms;
using MRP.Core;
using MRP.Entities;
using Telerik.WinControls.UI;

namespace MRP.GUI
{
    public partial class AssemblyForm : RadForm
    {
        private int _selectedAssemblyId;
```

```
private int _selectedRootComponentId;

public AssemblyForm()
{
    InitializeComponent();
    LoadAssemblies();
}

private void LoadAssemblies()
{
    using (var db = new DataContext())
    {
                        var assemblies = db.Assemblies.Include(x =>
     x.StartComponent).ToList();
        if (assemblies.Count > 0)
            foreach (var x in assemblies)
            {
                listView1.Items.Add(new ListViewItem {
                    Text = x.Name,
                    Tag = new SelectedAssembly
                    {
                        AssemblyId = x.Id,
                        RootComponentId = x.StartComponent.Id
                    }
                });
            }
    }
```

```
}

private void ClearAssemblies()
{
   listView1.Clear();
}

private void CreateTree()
{
   using (var db = new DataContext())
   {
      var treeComponent = db.Components
         .Include(x => x.Children.Select(
            y => y.Children.Select(
               z => z.Children.Select(
                  e => e.Children.Select(p => p.Children)))
         ))
         .FirstOrDefault(x => x.Id == _selectedRootComponentId);

      var node = new RadTreeNode {Value = null, Expanded = true, Text
   = "Root"};
      radTreeView1.Nodes.Add(node);
      CreateLeafs(node, treeComponent);
   }
}

      private void CreateLeafs(RadTreeNode parentNode, Component
   component)
```

```
{
   var componentNode = new RadTreeNode(component.Name) {Value =
    component, Expanded = true};
   parentNode.Nodes.Add(componentNode);
   if (component.Children != null)
      foreach (var children in component.Children)
         CreateLeafs(componentNode, children);
}

private void AddComponent_Click(object sender, EventArgs e)
{
   if (radTreeView1.SelectedNode?.Value == null) return;

   var addComponentForm = new AddComponentForm();
                        addComponentForm.OnAddNewComponent +=
    OnAddNewComponent;
   addComponentForm.Show();
}

                private void OnAddNewComponent(object sender,
    AddComponentEventArgs e)
{
   var component = (Component) radTreeView1.SelectedNode.Value;
   using (var db = new DataContext())
   {
                           var parent = db.Components.Include(x =>
    x.Children).FirstOrDefault(x => x.Id == component.Id);
      if (parent != null)
```

```
        {
            var newComponent = new Component
                                {Name = e.ComponentName, AssemblyId =
    _selectedAssemblyId, Unit = "шт"};
            parent.Children.Add(newComponent);
            db.SaveChanges();
            ClearTree();
            CreateTree();
        }
        else
        {
            MessageBox.Show(@"Компонент не найден");
        }
    }
}

private void ClearTree()
{
    radTreeView1.DataSource = null;
    radTreeView1.Nodes.Clear();
}

private void DeleteComponent_Click(object sender, EventArgs e)
{
    var component = (Component) radTreeView1.SelectedNode?.Value;
    if (component?.Parent == null) return;
    using (var db = new DataContext())
```

```
    {
                                var parent = db.Components.Include(x =>
     x.Children).FirstOrDefault(x => x.Id == component.Id);
        if (parent != null)
        {
            var treeComponents = Utils.GetTreeComponents(parent);
            treeComponents.Sort((a, b) => a.Id > b.Id ? 1 : -1);
            db.Components.RemoveRange(treeComponents);
            db.SaveChanges();
            ClearTree();
            CreateTree();
        }
        else
        {
            MessageBox.Show(@"Компонент не найден");
        }
    }
}


private void AddAssemblyBtn_Click(object sender, EventArgs e)
{
    var addAssemblyForm = new AddAssemblyForm();
    addAssemblyForm.OnAddNewAssembly += (sender1, e1) =>
    {
        AddNewAssembly(e1.AssemblyName);
        ClearAssemblies();
```

```
            LoadAssemblies();
            ClearTree();
        };
        addAssemblyForm.Show();
    }

    private void AddNewAssembly(string name)
    {
        using (var db = new DataContext())
        {
            var specification = new Assembly
            {
                Name = name
            };

            db.Assemblies.Add(specification);
            db.SaveChanges();

            var rootComponent = new Component
            {
                Name = name,
                Unit = "шт",
                AssemblyId = specification.Id
            };

            specification.StartComponent = rootComponent;
```

```
            db.Entry(specification).State = EntityState.Modified;
            db.SaveChanges();

            var mainPlan = new MainPlan {Assembly = specification};

            db.MainPlans.Add(mainPlan);
            db.SaveChanges();
        }
    }


     private void AssembliesList_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        if (listView1.SelectedItems.Count <= 0) return;

                                var selectedAssembly = (SelectedAssembly)
    listView1.SelectedItems[0].Tag;
        _selectedAssemblyId = selectedAssembly.AssemblyId;
        _selectedRootComponentId = selectedAssembly.RootComponentId;
        ClearTree();
        CreateTree();
    }

    private void AssemblyForm_Load(object sender, EventArgs e)
    {

    }
```

```
        private void radTreeView1_SelectedNodeChanged(object sender,
    RadTreeViewEventArgs e)
{

    var component = (Component)e.Node?.Value;
    if (component == null)
    {
        radLabel4.Text = "";
        radLabel5.Text = "";
        radLabel6.Text = "";
        return;
    }

    radLabel4.Text = component.ExecutionTime.ToString();
    radLabel5.Text = component.CountInStore.ToString();
    radLabel6.Text = component.Volume;
}

private void DeleteAssemblyBtn_Click(object sender, EventArgs e)
{
    using (var db = new DataContext())
    {
        var rootComponent = db.Components.FirstOrDefault(x => x.Parent
    == null && x.AssemblyId == _selectedAssemblyId);

        Utils.DeleteComponentTree(rootComponent.AssemblyId.Value,db)
    ;
```

```
            var plan = db.MainPlans.Include(x => x.Assembly).FirstOrDefault(x
        => x.Assembly.Id == _selectedAssemblyId);
            db.MainPlans.Remove(plan);
            db.SaveChanges();

                var assembly = db.Assemblies.FirstOrDefault(x => x.Id ==
        _selectedAssemblyId);
            db.Assemblies.Remove(assembly);
            db.SaveChanges();

            db.Components.Remove(rootComponent);
            db.SaveChanges();
        }

        ClearAssemblies();
        ClearTree();

        LoadAssemblies();
    }
  }

  public class SelectedAssembly
  {
    public int RootComponentId { get; set; }
    public int AssemblyId { get; set; }
  }
}
```

9. Добавление компонента в форму

```
using System;
using System.Windows.Forms;

namespace MRP.GUI
{
    public partial class AddComponentForm : Telerik.WinControls.UI.RadForm
    {
        public EventHandler<AddComponentEventArgs> OnAddNewComponent { get; set; }
        public AddComponentForm()
        {
            InitializeComponent();
        }

        private void radButton1_Click(object sender, EventArgs e)
        {
            if (radTextBox1.Text.Length == 0)
            {
                MessageBox.Show(@"Введите название компонента");
                return;
            }

            OnAddNewComponent?.Invoke(this, new AddComponentEventArgs
            {
```

```
                ComponentName = radTextBox1.Text
            });
            Close();
        }
    }

    public class AddComponentEventArgs : EventArgs
    {
        public string ComponentName { get; set; }
    }
}
```